



UNIVERSITAT POLITÈCNICA DE CATALUNYA

Sensorització d'un establiment i obtenció de les dades a través de l'entorn web.

7 de juliol de 2016

Dissertació que presenta XAVI LLORDELLA I VILLARÓ
sota la direcció de DR. FRANCISCO DEL ÁGUILA LÓPEZ
per assolir el grau d'Enginyer en Sistemes TIC.

Aquesta obra està subjecta a una llicència Attribution-NonCommercial-ShareAlike 3.0 Spain de Creative Commons. Per veure'n una còpia, visiteu <http://creativecommons.org/licenses/by-nc-sa/3.0/es> o envieu una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Índex

Agraïments	vii
Abstract	ix
Resum	xi
1 Introducció	1
2 Situació actual	3
2.1 Millores proposades	3
3 Descripció i objectius del projecte	5
3.1 Descripció del projecte	5
3.2 Objectius	6
4 Elements i mòduls del sistema	7
4.1 Introducció	7
4.2 Components del projecte	7
4.2.1 Raspberry	7
4.2.2 DS18D20	8
4.2.3 CY-191B-P-Y	9
4.2.4 Foscam C1	10
4.2.5 Commutador de xarxa	11
4.3 Relació entre els components	12
4.4 Mòduls del sistema	12
5 La pàgina web	15
5.1 Introducció	15
5.2 Decisió del framework	15
5.3 Disseny web	16
5.4 Desenvolupament	16
5.4.1 Menú i plantilla	16
5.4.2 Pàgina simple amb informació	18
5.4.3 Formulari de contacte	19
5.4.4 Pestanya de productes	19
5.4.5 Apartat d'administració	22
6 Configuració dels mòduls del sistema	25
6.1 Introducció	25
6.2 Mòdul de temperatura	25
6.2.1 Hardware	25

6.2.2	Software	27
6.2.3	Problemes	27
6.2.4	Conclusions mòdul	29
6.3	Mòdul reflector	29
6.3.1	Hardware	30
6.3.2	Software	31
6.3.3	Problemes	34
6.3.4	Conclusions mòdul	34
6.4	Mòdul càmera	35
6.4.1	Hardware	35
6.4.2	Software	35
6.5	Mòdul servidor raspberry	35
6.5.1	Preparació inicial de la raspberry	35
6.5.2	Servidor apache per servir web2py	37
7	Implementació del sistema final	39
7.1	Introducció	39
7.1.1	Instal·lació mòduls de temperatura	39
7.1.2	Instal·lació mòduls reflectors	41
7.1.3	Instal·lació mòdul càmera	44
7.1.4	Instal·lació mòdul servidor	44
7.2	Sistema final	44
7.3	LOPD - Llei Orgànica de Protecció de Dades	44
8	Integració del sistema final a la web	47
8.1	Integració del mòdul temperatura	47
8.1.1	Sensor DS18B20 2	47
8.1.2	Sensor DS18B20 1 i 3	50
8.2	Integració del mòdul reflector	52
8.3	Integració del mòdul càmera	53
9	Conclusions	59
10	Futures millores	61
10.1	Millores en la pàgina web	61
10.2	Millores en els sensors	61

Índex de taules

4.1	Taula de característiques de les raspberry utilitzades en el projecte. [8], [17]	8
4.2	Taula de característiques del sensor CY-191B-P-Y	10
4.3	Taula de característiques de foscarn C1. [9]	11
6.1	Taula de connexions del sensor DS18B20 i el GPIO de la raspberry.	26
6.2	Taula de connexions del sensor DS18B20 i el GPIO de la raspberry.	30
6.3	Taula de la ubicació dels sensors a la porta corresponent.	31

Índex de figures

4.1	Raspberry Pi 2 i 3 Model B	8
4.2	GPIO de la raspberry [18]	9
4.3	Sensor DS18B20	9
4.4	Sensor reflector no escollit.	10
4.5	Sensor CY-191B-P-Y	10
4.6	Càmera foscaml C1	11
4.7	Commutador de xarxa TP-LINK 5 ports	12
5.1	Captura de l'inici del lloc web www.laquitxalla.com	17
5.2	Vista <i>productes/cotxets</i> de la pàgina web	22
5.3	Captura del peu de pàgina.	22
5.4	Apartat d'administració del lloc web	23
5.5	Taules de la base de dades	23
5.6	Formulari per a afegir un nou producte a la web.	24
5.7	Formulari per modificar un producte existent a la web.	24
6.1	Pins del sensor DS18B20	26
6.2	Connexions DS18B20 amb la raspberry.	26
6.3	Esquema connexions DS18B20, i resistor $4.7K\Omega$	27
6.4	Lectura del fitxer <code>w1_slave</code> , el qual ens mostra la temperatura actual 27.625	27
6.5	Codi que ens retorna constantment la temperatura actual, Font: [14]	28
6.6	Execució del fitxer <code>thermometer.py</code> amb el codi 6.5.	29
6.7	Plànol de l'entrada i ubicació dels sensors reflectors	30
6.8	Connexions pots sensor CY-191B-P-Y.	30
6.9	Màquina d'estats corresponent a la gestió dels sensors reflectors (<code>reflective.py</code>).	32
6.10	Captura del navegador a l'hora de configurar la càmera Foscam C1	35
6.11	Captura de l'aplicació del mòbil a l'apartat de configuració	36
7.1	Ubicació dels diferents components del sistema final en la botiga.	40
7.2	Sensor de temperatura 1: Ubicació i visió final d'aquest.	41
7.3	Sensor de temperatura 2: Ubicació i visió final d'aquest.	41
7.4	Sensor de temperatura 3: Ubicació i visió final d'aquest.	42
7.5	Sensor reflector 1: Ubicació i visió final d'aquest.	42
7.6	Sensor de reflector 2: Ubicació i visió final d'aquest.	43
7.7	Sensor reflector 3: Ubicació i visió final d'aquest.	43
7.8	Càmera: Ubicació i visió final d'aquesta.	44
7.9	Raspberry servidor: Ubicació i visió final d'aquesta.	45
7.10	Interconnexions de la infraestructura completa del projecte.	45
8.1	Part de la funció <code>temp()</code> del controlador <code>default.py</code>	48

8.2	Script encarregat de consultar la nova dada del fitxer <code>temp.dat</code>	49
8.3	Scripts encarregats de cridar la funció <code>temperature()</code> cada minut.	49
8.4	Divisió HTML on s'identifica l'element <code>id="sensor2"</code> , el qual serà actualitzat amb javascript.	49
8.5	Captura de pantalla de la informació de temperatura plasmada a la web.	52
8.6	Captura de pantalla de la informació corresponent als clients actuals.	53
8.7	Captura del pas 1 de la transmissió del vídeo en protocol HTTP	55
8.8	Captura del pas 2 de la transmissió del vídeo en protocol HTTP	56
8.9	Captura del pas 3 de la transmissió del vídeo en protocol HTTP	56
8.10	Errors en l'execució de la comanda per transmetre el vídeo a la raspberry.	57
8.11	Missatge previ a l'execució de la comanda per veure el vídeo en local.	57
8.12	Captura de pantalla de la visió de la càmera des de l'aplicació local.	58

Agraïments

Aquest treball ha estat realitzat a la botiga dels meus pares, per tant la primera i la més especial menció ha de ser per a ells. A ells els dec el projecte per deixar-me gaudir inicialment fent proves amb els sensors a la botiga i més endavant, i ja immers dins del projecte, la instal·lació de tots els microcontroladors i sensors necessaris.

També mereix especial atenció el tutor del meu treball, F. del Águila López, per assessorar i ajudar-me en tot moment en tots els punts del projecte.

A part també agrair al Xavi Roig el temps destinat a dissenyar i imprimir les peces que subjectaran els sensors foto reflectors a les portes de la botiga.

Finalment agrair a la Jèssica Pallarès per ajudar-me en les fotos dels productes de la pàgina web, i a ella, els meus pares, la meva germana i a la resta de la família per donar-me suport en tot moment durant el transcurs del treball.

Abstract

Constant changing and capacity to modernize ourselves is becoming essential in our age. Because of this, and taking advantage of owning a family business in a constant process of modernization, we have studied the lacks and possible improvements of this business, which have been implemented in a web environment.

The current thesis describes the study of the improvements made as well as the solutions chosen for each case. The chosen solutions have been the creation of a web page, the implementation of smart modules (temperature and reflector sensors), and the installation of a security camera. Then, each module that is going to be a part of the final system is explained. Lastly, the final system, once it has been implemented and it is working correctly there will be a server (raspberry) which will acquire data from the modules and it will display this data on the web page of the business.

Once the project has been finished we will be able to see the improvements as well as the future lines of work to continuing to make the system even more robust, complete and generic.

Resum

El canvi constant i la capacitat de modernitzar-nos està essent imprescindible en l'era actual. Per aquest motiu, i aprofitant el fet de tenir un negoci familiar amb procés de modernització, s'han estudiat les mancances i possibles millores d'aquest, i s'han implementat i plasmat a través d'un entorn web.

En el present treball, es descriu l'estudi de les millores realitzat, així com les solucions escollides per a cada cas. Les solucions escollides han estat la creació d'una pàgina web, la creació d'uns mòduls intel·ligents (temperatura i sensors reflectors), i la instal·lació d'una càmera de seguretat. A continuació, i no menys important, s'explica cada mòdul que formarà part del sistema final, i per acabar, el sistema final. Pel que fa al sistema final, una vegada implementat i funcionant correctament, hi haurà un servidor (raspberrypi) el qual adquirirà dades dels altres mòduls i les plasmarà a la pàgina web de la botiga.

Una vegada finalitzat el projecte podrem veure el seguit de millores i futures línies a seguir per continuar fent el sistema més robust, complet i genèric.

1 Introducció

Cada dia estem en una societat més automatitzada, comunicada, controlada i còmoda per als éssers humans. Hi ha moltes aplicacions que ens fan la feina que abans havíem de fer manualment, també hi ha aparells que pensen i fins i tot actuen per nosaltres, i cada dia més, es controla tot el que fem.

La part de controlar el que fem encara està malvista per una part de la societat, no ens agrada que en tot moment gràcies als mòbils o altres aparells intel·ligents sàpiguem on som i que fem, contradictòriament però, estem penjant informació i fotografies del nostre dia a dia. Amb això vull dir que ja ens hem acostumat també a saber i voler saber com més coses millor i per tant a gaudir d'alguns dels avantatges que les noves tecnologies ens ofereixen.

En les grans empreses i ja en les petites també, s'està utilitzant una àmplia gamma de productes que faciliten dades, estadístiques o informació de qualsevol tipus que pugui ajudar a millorar o facilitar la feina dels treballadors. Aprofundint en aquest àmbit i aprofitant de tenir un negoci familiar, s'ha volgut estudiar les necessitats o possibles millores d'aquest, i actuar conseqüentment.

Per tant l'objectiu d'aquest treball anirà enfocat a millorar un establiment afegint un seguit de funcionalitats i comoditats, mitjançant sensors per tal d'ajudar als treballadors en el seu dia a dia al treball. A més a més, es dissenyarà una pàgina web que a part de servir per donar a conèixer la botiga, servirà per plasmar la informació dels sensors en tot moment.

L'abast del projecte està fitat a l'estudi de les necessitats de l'establiment, així com l'elecció de la millor solució per a les necessitats escollides. A continuació s'estudiarà cada solució per separat per tal de conèixer millor com tractar amb el sensor, aparell, càmera o del que es tracti cada solució. Per acabar, es configurarà, instal·larà i integraran les resolucions escollides i se'n provarà la funcionalitat total. A part es pretén proposar un seguit de millores que puguin ampliar encara més els temes desenvolupats en el present projecte.

Pel que fa a l'estructuració del projecte, s'ha decidit incorporar-hi una descripció de tot el material i procediments usats per tal que la resta del projecte sigui perfectament comprensible per al lector. Seguidament ens torbarem amb les primeres proves amb els sensors i les primeres configuracions per tal de fer funcionar els sensors i aparells. Just després s'explicarà la implementació total de cada una de les solucions així com els problemes sorgits durant el transcurs del projecte, i la integració de la informació al lloc web. Finalment, es troben les conclusions i possibles millores, i ja que el projecte ha estat desenvolupat en un entorn familiar, de ben segur que moltes de les millores proposades es desenvoluparan una vegada finalitzat el projecte. Per a la realització d'aquest treball s'ha seguit un ordre cronològic, ja que quasi bé totes les tasques estaven lligades a procediments anteriors. La informació i documentació necessària pel treball ha estat consultada en la seva totalitat mitjançant internet.

2 Situació actual

Actualment l'empresa està en creixement i es pretén desenvolupar un seguit de millores que ajudin a controlar diferents aspectes per fer el dia a dia de la botiga més còmode i fàcil. Com ja s'ha explicat anteriorment s'ha estudiat quines eren les necessitats i millores aplicables a la botiga. La botiga és de puericultura i roba des del néixer fins a 2 anys, annexa hi ha una altra part de la botiga on hi ha roba des dels 2 anys fins a adult.

L'establiment ha estat reformat fa poques setmanes, i ha canviat la imatge corporativa, i es pretén canviar-ne el mobiliari en poc temps. D'ençà d'aquest canvi, ja s'havia proposat crear-ne una pàgina web, però fins al moment no s'havia dut a terme. Per tant s'ha aprofitat el projecte i la motivació per a modernitzar la botiga per tal de crear-ne la pàgina web com a extra a la instal·lació més tècnica que té com a objectiu principal aquest projecte.

Seguint el mateix fil, s'ha pensat en la seguretat de la botiga, instal·lant càmeres de seguretat per a tenir en tot moment la visió de l'interior del negoci.

A més hi ha molt aparador que fa que la temperatura canviï molt durant el dia, depenent de si és al matí o a la tarda, o també de l'estació de l'any. Per això s'ha pensat instal·lar uns sensors de temperatura per tenir constància de les variacions de temperatura en tot moment.

Cal mencionar també que a la botiga hi ha dues treballadores, la qual una d'elles és la gerent. Sovint la treballadora és la que està físicament a la botiga i la gerent està a casa o despatx treballant amb els temes administratius o realitzant les compres als proveïdors. Constantment la treballadora ha d'estar trucant o avisant a la gerent quan la botiga s'emplena i és necessari una segona dependent. Tenint en compte aquesta realitat, s'ha volgut actuar instal·lant i muntant un sistema que permeti tenir constància i controlar de forma automàtica la quantitat de persones que hi ha en l'establiment.

2.1 Millores proposades

Una vegada explicada la situació actual i vistes quines seran les solucions o millores que formaran el projecte, ens disposem a explicar més acuradament cada una d'elles.

- **Pàgina web:**

Com ja s'ha dit anteriorment, la idea de la pàgina web no és nova, sinó que des que s'ha canviat la imatge corporativa es volia disposar de pàgina web. Així doncs i aprofitant el projecte es dissenyarà i crearà una pàgina web per tal de donar a conèixer la botiga, i per tant augmentar el seu target. A part d'això hi haurà un apartat d'administració des del qual només es podrà accedir si prèviament en disposes l'accés.

En l'apartat d'administració es podrà veure la informació referent als sensors instal·lats, temperatura de tots els punts de la botiga, quants clients hi ha dins l'establiment, i per últim també es podrà gaudir una imatge en viu de la botiga.

- **Càmera de seguretat:**

La càmera de seguretat, serà una càmera IP de la marca Foscam i model C1 amb protocol RTSP per tal de poder accedir directament a ella i obtenir-ne el vídeo en [streaming](#). Aquesta anirà situada a l'extrem esquerre de la botiga per tal de poder gravar el màxim d'angle possible, i així veure com més tros millor. Mirant la Figura 7.10 ses pot entendre millora la ubicació de la càmera.

- **Sensor de temperatura:**

El sensor de temperatura serà el DS18B20, i se'n col·locaran tres, un a l'extrem esquerre de la botiga, un al pilar central, al costat del taulell on hi haurà majoritàriament els clients i l'últim a l'extrem esquerre del fons de la botiga.

Se'n posen 2 als extrems per tal de poder comparar com varia la temperatura en diferents indrets, un a la vora dels aparadors, l'altre al mig de l'establiment i per tant dels clients i l'últim en un indret on no toca mai el sol però ubicat al costat de la petita puericultura, és a dir que la gent també es mourà per aquí.

- **Sensor foto reflectiu:**

Per últim el sensor reflectiu serà el CY-191B-P-Y de la marca Panasonic, i se n'instal·laran 3, un a cada una de les 3 portes d'entrada a la botiga.

La botiga disposa d'una doble entrada, la porta que dóna al carrer i una vegada superada, hi ha dues portes una a mà dreta i l'altra tot recte que donen accés a la botiga de roba i a la de puericultura respectivament. En la figura 6.7 podem veure un plànol de l'entrada per aclarir aquest punt.

3 Descripció i objectius del projecte

3.1 Descripció del projecte

Una vegada s'han explicat els objectius, podem explicar més acuradament l'abast i funció final del present projecte. Com ja s'ha dit en els punts anteriors, l'objectiu final serà aconseguir dissenyar i programar una pàgina web per a la botiga amb la finalitat de donar a conèixer la botiga a internet, i informar del que oferim en ella. A més a més, es vol veure la informació dels sensors reflectors, temperatura i càmera, a través de la part d'administració de la web, totalment a temps real.

Anem a detallar una mica més que es vol incorporar en la pàgina web. Volem una pàgina web en la qual hi hagi tota la informació de la botiga, informació general, productes, història, localització, contacte, galeria d'imatges, ofertes, etc. En l'apartat dels productes es farà una classificació en categories, permetent així facilitar als usuaris la cerca d'un producte determinat. A part, tindrà un apartat d'administració amb les següents característiques:

- Només s'hi podrà accedir si disposes de les credencials d'accés.
- Es podrà recuperar la contrasenya en cas de pèrdua..
- Una vegada iniciada la sessió es podrà modificar el correu i nom d'usuari.
- Es podrà consultar la temperatura dels 3 sensors instal·lats a la botiga.
- Podrem saber quanta gent hi ha en l'establiment, gràcies als 3 sensors reflectors instal·lats a l'entrada.
- Tindrem accés en temps real a la càmera habilitada.
- Es podrà accedir a un segon menú des del qual podrem gestionar les categories i productes que apareixeran a la web.

Una vegada explicada la part de l'entorn web, s'explica la part de hardware. A continuació s'expliquen els 3 elements que s'han escollit per tal de millorar l'establiment. La càmera, escollida per a millorar la seguretat, estarà connectada a internet per tal de poder accedir-hi de forma ràpida i des de qualsevol lloc.

Pel que fa als sensors de temperatura, se n'instal·laran 3, com s'ha dit anteriorment. El fet d'instal·lar-ne 3, ens permetrà conèixer en quin indret fa més calor o fred, i actuar conseqüentment amb l'aire condicionat o calefacció. També podrem alterar lleugerament la temperatura dels punts més calents de la botiga amb les persianes, ja que és un local amb molt d'aparador, i per tant fa que el sol escalfi molt determinades zones.

Finalment, els sensors reflectors, s'instal·laran a les portes de la botiga. S'ha instal·lat sensors en les 3 portes, ja que ambdues botigues comuniquen interiorment i els clients poden entrar per la porta d'un establiment i sortir per la de l'altre. Per tant si es vol comptabilitzar els clients que hi ha dins la botiga, serà necessari el muntatge dels 3 sensors.

Tots els sensors seran connectats a una [raspberry](#) i aquesta a internet, per tal de poder gestionar la informació dels sensors des d'una altra raspberry que farà de servidor web i en la web serà on es plasmarà totes les dades obtingudes.

La raspberry servidor, serà l'encarregada d'obtenir les dades dels sensors i també de la càmera IP que com bé s'ha dit, també estarà connectada a la xarxa.

3.2 Objectius

Una vegada escollides les millores a realitzar podem dir quines són les ambicions i els objectius del present treball. Com bé s'ha explicat en les millores proposades, es vol aconseguir crear una pàgina web, amb un estil adequat per a una botiga de puericultura i roba, totalment genèrica, i per tant, amb la possibilitat d'incorporar altres funcionalitats de forma ràpida i senzilla. A més es vol implementar mòduls intel·ligents els quals a través del servidor es pugui consultar la informació desitjada. Finalment, la creació del servidor web amb una raspberry, i la consulta de la informació dels mòduls per adquirir la informació a temps real de cada sensor, és l'al·licient més gran del projecte. Una vegada aconseguit el que s'acaba de citar, s'intentarà pensar amb una línia de futures millores i un fil per a fer d'aquest projecte simplement l'inici d'un gran projecte.

4 Elements i mòduls del sistema

4.1 Introducció

En aquest apartat ens centrarem a descriure en què consistirà el projecte i a definir i explicar tots els components del projecte. Inicialment es donarà un llistat del material necessari i la informació teòrica dels microcontroladors i sensors per tal d'ajudar a tenir una millor comprensió a l'hora de llegir el projecte. Finalment es farà una breu explicació de quina relació tindran entre ells.

4.2 Components del projecte

- 2 Raspberry Pi 3 Model B
- 1 Raspberry Pi 2 Model B
- 3 Sensors de temperatura (DS18B20)
- 1 Càmera IP (Foscam C1)
- 3 Sensors reflectors (Panasonic CY-191B-P-Y)
- 3 plaques de coure verges
- 3 resistències de 3.6 k Ω
- 3 resistències de 4,7 k Ω
- 3 resistències de 10 k Ω
- Cable rígid
- Cable [ethernet](#)
- Commutador de xarxa ([Switch](#))

4.2.1 Raspberry

La Raspberry Pi [Figura 4.1] és un ordinador mono placa de baix cost, desenvolupat en el Regne Unit per la Fundació Raspberry Pi. L'objectiu principal d'aquest disseny és estimular l'ensenyança de les ciències de la computació, però també s'ha popularitzat com a plataforma per a dissenys d'aficionats i per a usos informàtics generals. [15]

En el treball s'utilitzarà la raspberry per tal de gestionar i poder comunicar-nos amb els sensors, i també per crear un servidor per poder allotjar la nostra pàgina web.

	Pi 2 Model B+	Pi 3 Model B+
Preu:	35 €	35 €
SoC:	Broadcom BCM2836	Broadcom BCM2837 1.2GHz
CPU:	ARM Cortex-A7, quad-core 900MHz	ARM Cortex-A53, 64bits, quad-core 1.2MHz
GPU:	VideoCore IV 250MHz	VideoCore IV 400MHz
Memòria (RAM):	1GB 450MHz	1GB 900MHz
PortsUSB 2.0:	4	4
Sortida de vídeo:	Connector RCA, HDMI	Connector RCA, HDMI
Sortida d'àudio:	Jack (3,5mm), 4 pols àudio i vídeo, HDMI	Jack (3,5mm), 4 pols àudio i vídeo, HDMI
Emmagatzematge integrat:	microSD	microSD
Connectivitat de xarxa:	10/100 Ethernet	10/100 Ethernet, WIFI i Bluetooth 4.1
Perifèrics de baix nivell:	40 x GPIO	40 xGPIO
Consum energètic:	600 mA (3.0 W)	600 mA (3.0 W)
Font d'alimentació:	5V (Micro-USB o GPIO)	5V (Micro-USB o GPIO)
Dimensions:	85mm x 56mm x 17mm	85mm x 56mm x 17mm
S.O. suportats:	Raspbian, Linux, Windows 10, ...	Raspbian, Linux, Windows 10, ...

Taula 4.1: Taula de característiques de les raspberry utilitzades en el projecte. [8], [17]



Figura 4.1: Raspberry Pi 2 i 3 Model B

GPIO de la raspberry

El GPIO (General Purpose Input/Output) [Figura 4.2] és un pin genèric programable per l'usuari en temps d'execució. No tenen cap propòsit especial definit ni s'utilitzen d'una forma predeterminada, sinó que la seva flexibilitat permet usar-los per a una gran varietat d'aplicacions diferents.[11]

En especial el GPIO de la raspberry és la interfície física entre la Raspberry Pi i el món exterior. En un nivell simple podem pensar que són interruptors que podem obrir i tancar. Es poden programar els pins per tal d'interactuar amb el món exterior, i els pins d'entrada no han de provenir d'un interruptor, sinó que poder fer-ho des d'un sensor o qualsevol altre senyal d'un ordinador o aparell. [12]

4.2.2 DS18D20

El DS18B20 1-wire [Figura 4.3] és un sensor de temperatura digital molt utilitzat en diferents aplicacions, controls termo-estàtics, sistemes industrials, termòmetres, sistemes tèrmicament sensibles, entre altres. A més la raspberry no disposa de ADC (Analogue to digital converter) per tant el fet d'utilitzar un sensor digital ens facilitarà molt la feina. Aquest tipus de sensors disposen d'una interfície de comunicacions d'un sol cable la qual en el nostre cas anirà

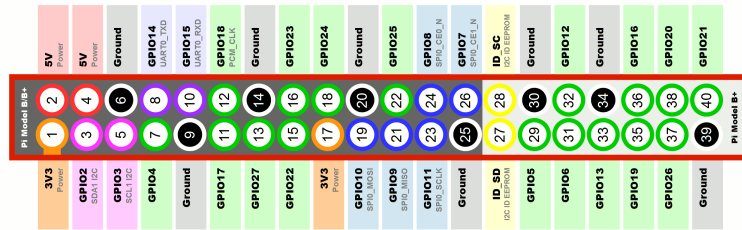


Figura 4.2: GPIO de la raspberry [18]

connectada al GPIO de la raspberry.[7]

A continuació podem veure les característiques principals d'aquest sensor:

- **Temperatura:** -55 a + 125°C
- **Temps conversió:** 93,75 ms (9 bits), 750 ms (12 bits)
- **Alimentació:** +3.0V a +5.5V DC o des de la línia de dades.
- **Altres:** Codi de sèrie de 64 bits guardat a la ROM

Aquest sensor és ideal per la raspberry, es comunica amb el microprocessador a través del (1-Wire bus), el qual només requereix una línia i terra (GND). A més aquest sensor pot derivar potència a través de la línia de dades ("Parasite Mode"), eliminant la necessitat d'una font de tensió externa.



Figura 4.3: Sensor DS18B20

4.2.3 CY-191B-P-Y

El sensor CY-191B-P-Y és un sensor destinat a la indústria per detectar el pas dels productes en una línia de producció. També s'utilitza en els ascensors i garatge per detectar el pas de persones i vehicles. De sensors foto-reflectors, n'hi ha de molts tipus. Hi ha els que necessiten un emissor i un receptor, i altres que només un emissor i un catadiòptric. Aquest darrer cas és l'utilitzat en el projecte, i l'emissor farà la funció de receptor també.

Inicialment s'havia mirat el sensor reflector que podem veure a la figura 4.4 però la distància de detecció era de l'ordre de pocs centímetres, i no ens servia per al que es volia. [19]



Figura 4.4: Sensor reflector no escollit.

Es necessitava que el sensor pogués detectar a una distància de 1.5 metres, ja que és la distància de la porta principal de l'establiment. Amb aquesta distància de detecció el preu dels sensors reflectors era molt elevat. A partir d'un rang de detecció de 50cm, el preu pujava dels 5€, al voltant dels 50€.

Finalment i assumint les despeses, el sensor foto-reflector escollit ha estat el CY-191B-P-Y [Figura 4.5], de la casa panasonic. Podem veure les característiques a la taula 4.2. [6] [4]

	CY-191B-P-Y
Preu:	45 €
Rang de detecció:	2m
Alimentació:	12V - 24V
Consum:	< 35mA
Tipus de sortida:	PNP, col·lector obert
Temps de resposta:	< 1ms

Taula 4.2: Taula de característiques del sensor CY-191B-P-Y



Figura 4.5: Sensor CY-191B-P-Y

4.2.4 Foscam C1

Una càmera IP és una videocàmera dissenyada especialment per enviar senyals de vídeo i en alguns casos d'àudio a través d'internet des d'un navegador o concentrador [hub](#) en una red

local LAN.

En les càmeres IP es poden integrar aplicacions de detecció de presència (inclús l'enviament de correu), gravació d'imatges o vídeos quan es detecti la presència, etc.

Utilitzarem una càmera IP per tal de poder veure tot el que passa a dins a la botiga en tot moment des del lloc web, com a medi de seguretat. La càmera escollida ha estat la Foscam C1 [Figura 4.6] amb les característiques més importants citades a la taula 4.3.

	Foscam C1
Preu:	69 €
Tipus de sensor:	1.4 " CMOS
Resolució de pantalla:	1.0 MegaPixels(1280*720)
Frame rate:	25fps(VGA) 23fps(720P)
Tipus de lent:	f:2.8mm,F:2.4
Angle de visió:	115°
Wireless:	IEEE 802.11b/g/n
Wireless security:	WEP,WPA,WPA2
format de video:	H.264
Navegadors suportats:	IE8, Firefox, Chrome, Safari
Alimentació:	5V/2A
Consum energètic:	< 3.0W

Taula 4.3: Taula de característiques de foscam C1. [9]



Figura 4.6: Càmera foscam C1

4.2.5 Commutador de xarxa

El commutador o switch (en anglès) [Figura 4.7] és un dispositiu electrònic on es connecten els ordinadors o diferents segments per formar una xarxa de molta més extensió, capacitat i eficiència que un concentrador. Optimitza el tràfic de dades de la xarxa de forma intel·ligent, és a dir, s'adapta a les velocitats màximes estàndards de cada dispositiu connectat. Per tant s'utilitzarà quan vulguem connectar múltiples xarxes, fusionant-les en una sola.

És un aparell molt semblant al [hub](#) , però té una gran diferència: Aquest sí que diferencia els equips connectats a ell, a través de la seva adreça MAC o IP . Les dades enviades per un ordinador arriben només a l'ordinador al qual s'ha enviat, creant una mena de canal de comunicació exclusiva entre l'origen i el destí.[3]



Figura 4.7: Commutador de xarxa TP-LINK 5 ports

4.3 Relació entre els components

Una vegada explicats els components utilitzats, s'ha cregut convenient explicar la relació entre ells. La càmera estarà connectada mitjanant un cable de xarxa (cable [ethernet](#)) al commutador de xarxa ([switch](#)), i aquest al [router](#), i per tant a internet. El sensor de temperatura (DS18B20) estarà unit a la [raspberry](#), i aquesta al commutador i a internet. Pel que fa als sensors reflectors, tindran la mateixa configuració que els sensors de temperatura.

Més endavant s'explica la configuració detallada de cada sensor i per tant de cada mòdul que constituirà el sistema final.

4.4 Mòduls del sistema

En aquest projecte es vol crear uns mòduls per tal d'aconseguir fer una infraestructura el més genèrica possible. Per aquesta finalitat, tindrem 4 mòduls disponibles, amb l'explicació completa de configuració de cada un, per tal de que puguem agafar els mòduls que vulguem de cada tipus i així poder tenir una arquitectura completament al nostre gust i adaptada a les necessitats de l'entorn desitjat.

A més, parlar amb nom de mòduls i no de components per separat, servirà per facilitar l'entesa al lector al llarg del projecte i així poder citar el conjunt d'elements que realitzen una funció concreta, simplement amb el nom del mòdul.

Els mòduls que disposarem i que utilitzarem per aconseguir la infraestructura del sistema final desitjat seran:

- **Mòdul de temperatura:** Aquest mòdul tindrà la funció de donar a conèixer la temperatura a temps real de l'indret on sigui instal·lat.
- **Mòdul reflector:** El mòdul reflector, detectarà el pas dels objectes o persones a temps real, contant quantes són les persones que han travessat el raig de llum emesa pel sensor. Si es vol obtenir el nombre de persones que entren o surten d'un local, com és el cas que ens ocupa, seran necessaris com a mínim dos mòduls reflectors.
- **Mòdul càmera:** El mòdul càmera, consta només d'una càmera IP configurada com s'explica en el següent apartat i simplement connectada a internet, per tal d'obtenir el vídeo a temps real a través de la xarxa.
- **Mòdul servidor web:** Per últim el mòdul servidor, encarregat d'allotjar la pàgina web i el servidor, [apache](#) en el present projecte, servirà també per obtenir dades de tots els altres mòduls instal·lats en el sistema final. Aquesta informació, podrà ser plasmada a la web, o bé fer-ne l'ús que es desitgi en cada cas.

5 La pàgina web

5.1 Introducció

Actualment la pàgina web és un element importantíssim per a un negoci. En aquesta societat, es busca molta informació per internet, i com més informació tingui la web millor. Per aquest fet i els explicats en la introducció del treball s'ha decidit crear una pàgina web.

El primer que necessitàvem saber per iniciar el disseny i programació de la web, era que volíem que tingués la web. Es volia que tingués un estil elegant i que els colors i l'estil estiguessin d'acord amb els productes i serveix que ofereix la botiga, en aquest cas, puericultura i roba de nadó. També es volia disposar de diversos apartats en la web, informació general, història, contacte, productes, multimèdia.

En l'apartat de productes hi haurà moltes fotos que plasmaran tots els productes disponibles a la botiga. Aquests productes canvien molt sovint, i per tant hem d'aconseguir que també siguin fàcils de modificar en el lloc web. Pel que fa a l'apartat de contacte, també s'ha de poder crear formularis, i enviar-los per correu.

A continuació s'explica la decisió del [framework](#) en funció de les característiques necessàries, el disseny de la web i la programació d'aquesta.

5.2 Decisió del framework

Per a la programació de la pàgina web, s'ha utilitzat HTML, CSS i Javascript, però també es va decidir utilitzar frameworks que facilitin la programació.

Inicialment es volia utilitzar un [framework backend](#), per a la gestió de la base de dades, autenticació i administració, com laravel o django [13] [20].

De laravel ja se'n coneixia el funcionament, ja que prèviament i en l'àmbit personal, s'havia utilitzat per a la realització d'una altra web. Utilitza el llenguatge PHP. Django en canvi, utilitza el llenguatge python, i s'havia utilitzat en una pràctica a la universitat.

Després es va proposar la utilització del framework web2py [24], utilitza python i és molt semblant a django. Per tant es va optar per fer una comparativa completa [21] [22], i en veure que eren quasi iguals es va escollir **web2py**, ja que és una proposta que utilitza python, llenguatge molt utilitzat durant el grau, i així s'aprenia un nou framework, idea molt motivadora per ampliar els coneixements en els temes web.

El framework escollit, és frontend i backend a la vegada, ja que web2py incorpora bootstrap [2] per al fronttend. És a dir, en la part de la base de dades, administració i autenticació treballa amb python i en la part d'estil, HTML i CSS, utilitza bootstrap.

S'aprofita aquest punt per mencionar també, que personalment, s'havia treballat amb un altre framework de frontend, dissenyat amb CSS i javascript anomenat materialize [5]. Aquest fra-

mework està basat en l'estil [minimalize](#) que utilitza google i incorpora certs components molt interessants i amb unes característiques molt apropiades sobretot per a l'apartat de *productes* de la web.

Utilitzar dos frameworks en una mateixa web, requereix certa experiència en disseny web, ja que tots incorporen unes característiques bàsiques i al incloure'n dos a la vegada, molts dels estils no quedaven bé, ja que hi havia conflicte a l'hora d'agafar l'estil definit en el codi HTML. Per tal de solucionar aquest problema, es va crear un full d'estil propi el qual vam anomenar `materialize_quitxalla.css` on es van incloure les definicions necessàries per a utilitzar el component desitjat. També va ser necessària crear un altre arxiu amb el javascript corresponent.

5.3 Disseny web

El disseny de la web, colors, estil, logotip, etc. va ser escollit conjuntament amb els propietaris de la botiga, ja que la idea ja la tenien, i es volia seguir amb el mateix estil que la Q Jove, botiga de roba juvenil. Per tant, es va dissenyar el logotip amb la mateixa lletra i estil, simplement modificant-ne el color.

El disseny de la resta de la web havia d'estar relacionat amb els colors de la imatge corporativa, i havia de ser amb uns tons suaus, entenent que la web anava adreçada a mares i pares amb fills d'edats entre 0 i 2 anys.

Finalment va quedar tal com ens mostra la figura 5.1. Podem veure més imatges de la web en els annexos.

5.4 Desenvolupament

Web2py és un framework de desenvolupament de codi lliure, per al desenvolupament àgil d'aplicacions web segures amb base de dades. Escrit i programable amb python. És totalment complet, ja que disposa de tots els components necessaris per construir aplicacions web totalment funcionals. Utilitza el patró [MVC](#), separa la representació de les dades (el model), de la presentació de les dades (la vista) i també de la lògica d'aplicació i flux de treball (el controlador). [24]

A continuació es parlarà de com s'han desenvolupat les diferents parts que corresponen al desenvolupament de l'entorn web.

5.4.1 Menú i plantilla

Per dissenyar menú, s'ha fet com veiem en el codi 5.1 en el model `menu.py`.

Codi 5.1: Codi corresponent a la definició del menú de la web

```
def _():
    app = request.application
    ctr = request.controller
    response.menu += [
        (T('Sobre Nosaltres'), False, URL('default', 'nosaltres')),
        (T('Productes'), False, '#', [
```

```

(T('Cotxets'), False,
  URL('default', 'productes/cotxets')),
(T('Cadires_passeig'), False,
  URL('default', 'productes/passeig')),
(T('Llits'), False,
  URL('default', 'productes/llits')),
(T('Parques'), False,
  URL('default', 'productes/parques')),
(T('Moises'), False,
  URL('default', 'productes/moises')),
(T('Cadires_cotxe'), False,
  URL('default', 'productes/cadires')),
(T('Trones'), False,
  URL('default', 'productes/trones')),
LI(_class="divider"),
(T('Petita_Puericultura'), False,
  URL('default', 'petita_puericultura')),
]),
(T('Galeria_imatges'), False, URL('default', 'media')),
(T('Contacte'), False, URL('default', 'contacte')),
]

```



Figura 5.1: Captura de l'inici del lloc web www.laquitxalla.com

Una vegada definit, es va definir un arxiu `layout.html` que serà la plantilla en la qual hi haurà el cap i el peu de la pàgina, i tots els altres documents ampliaran aquesta plantilla.

En el `layout.html` s'ha definit el logotip inicial, el menú i el peu de pàgina, ja que en totes les pàgines de la web, hi haurà el mateix cap i peu de pàgina. Veiem en el codi 5.2 com s'ha incorporat el menú en la plantilla base.

Codi 5.2: Codi corresponent a la crida del menú en la plantilla

```
<nav class="navbar navbar-custom">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-
        toggle="collapse" data-target="#menu-quitxalla">
        <i class="material-icons">dehaze</i>
      </button>
      <a class="navbar-brand hidden-lg hidden-md" href="#"> La
        Quitxalla </a>
    </div>
    <div class="navbar-collapse collapse" id="menu-quitxalla">
      <ul class="nav navbar-nav">
        {{ if response.menu: }}
        {{=MENU(response.menu, _class='nav navbar-nav', li_class='
          dropdown', ul_class='dropdown-menu') }}
        {{ pass }}
        {{='auth' in globals() and auth.navbar('Welcome', mode='
          dropdown') or ''}}
      </ul>
    </div>
  </div>
</nav>
```

5.4.2 Pàgina simple amb informació

Per a la creació d'una pàgina simple d'informació, el que s'ha de fer és crear una funció corresponent a la vista, en el controlador amb el nom corresponent al nom de la carpeta on estarà ubicat l'arxiu.

Per tant si volem crear la pàgina, *història*, haurem de crear un fitxer `historia.html` dins del directori `views/default` i definir tal com es veu en el codi 5.3, tota la informació que necessitem utilitzar per a la creació de la vista. En aquest cas, i si el text de la història no es modifica mai, per tant no està a la base de dades, la vista definida serà buida. En l'apartat 5.4.4 podrem veure una vista que accedeix a la base de dades.

Codi 5.3: Codi corresponent a una vista que retorna un diccionari buit.

```
def historia():
    return dict()
```


5.4.3 Formulari de contacte

Per a la creació del formulari de contacte, primer s'ha definit una nova taula a la base de dades, fitxer `models/db.py`. [Codi 5.4]

Codi 5.4: Codi corresponent a la definició de la taula `contacte` a la base de dades.

```
db.define_table('contact',
    Field('nom', requires=IS_NOT_EMPTY()),
    Field('correu', requires=[IS_NOT_EMPTY(), IS_EMAIL()]),
    Field('telefon', requires=IS_NOT_EMPTY()),
    Field('missatge', 'text', requires=IS_NOT_EMPTY())
)
```

A continuació en el controlador, hem definit una vista que agafa els camps entrats a la base de dades, i els plasma a la vista `contacte` en forma de formulari. [Codi 5.5]

Codi 5.5: Codi corresponent a la funció `contacte` del controlador `default.py`

```
def contacte():
    form=SQLFORM.factory(db.contact)
    if form.accepts(request):
        response.flash = 'Mail_sent!'
        # Send Email
        mail.send(to=['info@laquitxalla.com'], subject='Consulta Web-_' +
            request.vars.name, message='Correu:_' + request.vars.mail + '\n'
            'Telefon:_' + request.vars.telf + '\nAssumpte:_' + request.vars.
            message)
    return dict(form=form)
```

Finalment en la vista `views/default/contacte.html` només va ser necessari el següent:

```
{{=form}}
```

5.4.4 Pestanya de productes

Pel que fa a la pestanya de productes, el procediment va ser exactament el mateix que el del disseny del formulari de contacte, amb la diferència que en el controlador, va ser necessari consultar i fer cerques condicionades a la base de dades, per tal de seleccionar els productes corresponents.

De productes n'hi ha de dos tipus: productes normals, amb un nom, model, marca, descripció, característiques i imatge, i altres, que només el nom, marca, color i imatge són camps obligatoris. Els primers productes citats són els corresponents a la puericultura (cotxets, llits, trones, etc.), els segons, corresponen a la petita puericultura (joguines, pitets, vaixelles, reposacaps, etc.).

Pel que fa a la base de dades, hi ha quatre taules corresponent als productes (Categories de puericultura, productes de puericultura, categories de petita puericultura i productes de petita puericultura). Per tant veiem que es podran afegir/treure categories tant de la puericultura en general, com de la petita puericultura. Podem veure com està definida la taula que ens permet afegir un producte en el codi 5.6.

Codi 5.6: Codi corresponent a la definició de la taula producte a la base de dades.

```
db.define_table('productes',
    Field('producte', requires=IS_IN_SET(['Cotxets', 'Cadires_de_
        passeig', 'Llits', 'Parques', 'Moises', 'Cadires_de_cotxe', 'Trones',
        'Petita_puericultura'])),
    Field('marca', requires=IS_NOT_EMPTY()),
    Field('model', requires=IS_NOT_EMPTY()),
    Field('color', requires=IS_NOT_EMPTY()),
    Field('descripcio', 'text', requires=IS_NOT_EMPTY()),
    Field('caracteristiques', 'text', requires=IS_NOT_EMPTY()),
    Field('imatge', 'upload', requires=IS_NOT_EMPTY())
)
```

Si ens fixem en el codi 5.1, es veu que la *url* del desplegable de productes, és *productes/cotxets*, per exemple. Això ens servirà per tal de què només calgui crear una vista i per tant un arxiu anomenat *views/default/productes.html* i automàticament es creïn les vistes corresponents a cada categoria de productes només clicant en el menú. Per tant en el codi 5.7 veiem que per mostrar els productes d'acord amb la *url* corresponent a la categoria clicada. Si en en menú de la pàgina web es clica sobre COTXETS, la *url* serà */default/productes/cotxets* per tant el controlador agafarà el nom *cotxets* i buscarà a la base de dades tots els productes que tinguin cotxet, com a categoria.

A continuació tota la informació referent al producte, (nom, model, marca, descripció, característiques i imatge), s'afegeix a una llista i aquesta es retorna per poder ser utilitzada en la vista.

Codi 5.7: Codi corresponent a la funció producte del controlador default.py

```
def productes():
    url = request.url.split("/")
    url = url[-1]
    for item in db(db.categories_puericultura.url == url).select(db.
        categories_puericultura.ALL):
        categoria = item.categoria
    productes=[]
    for item in db(db.productes.producte == categoria).select(db.
        productes.ALL):
        producte=[]
        producte.append(item.producte)
        producte.append(item.marca)
        producte.append(item.model)
        producte.append(item.color)
        producte.append(item.descripcio)
        producte.append(item.caracteristiques)
        producte.append(item.imatge)
        productes.append(producte)
    return dict(url=url, productes=productes, categoria=categoria)
```

Finalment només queda tractar la informació que el controlador proporciona. El controlador dóna una llista, on cada element és una llista corresponent a tota la informació d'un producte.

Per aquest motiu, es recorre la llista de productes per obtenir la llista d'un sol producte, i recorrent la segona llista s'obté tota la informació de cada producte. Per plasmar cada camp de la informació referent a un producte en la vista es farà de la mateixa forma que s'ha fet en el formulari de contacte. [Codi 5.8]

Codi 5.8: Codi corresponent a la vista producte.

```
{{extend 'layout.html'}}

<div class="container">
  <h3 class="center_thin"> {{=categoria}} </h3>
  <div class="row">
    {{for producte, marca, model, color, descripcio, caracteristiques,
      imatge in productes:}}
    <div class="col-lg-3">
      <div class="card">
        <div class="card-image_waves-effect_waves-block_waves-light">
          >
          <img class="activator" src={{=URL('download', args=imatge)}}
            >
        </div>
        <div class="card-content">
          <span class="card-title_activator_grey-text_text-darken-4">{{=marca}}
            
          </span>
          <p>{{=model}} - {{=color}}</p>
        </div>
        <div class="card-reveal">
          <span class="card-title_grey-text_text-darken-4">{{=marca}}
            - {{=model}}
          
          </span>
          <p><b>Descripcio:</b><br>
            {{=descripcio}}<br><br>
            <b>Caracteristiques:</b><br>
            {{=caracteristiques}}
          </p>
        </div>
      </div>
    </div>
    {{pass}}
  </div>
<br>
</div>
```

Finalment us deixem en com és l'apartat **frontend** de la vista *productes/cotxets*. [Figura 5.2]



Figura 5.2: Vista *productes/cotxets* de la pàgina web

5.4.5 Apartat d'administració

En el peu de pàgina, [Figura 5.3] tenim un botó que permet l'accés a l'apartat d'administració, figura 5.4, on es veu l'accés a clients, temperatura i càmera. També es disposa d'un altre botó a sota que permetrà l'accés a afegir/modificar/eliminar nous productes o categories.



Figura 5.3: Captura del peu de pàgina.

En la figura 5.5 es veu com és l'apartat de modificació dels productes i en les figures 5.6 i 5.7 com es veu el formulari per afegir un nou producte i el d'edició d'un de ja existent.

Per veure el codi complet de la pàgina web o si desitgeu veure més imatges de la pàgina web, podeu trobar-ho en l'apartat d'annexes. A més podeu consultar el funcionament i les vistes de frontend a la web www.laquitxalla.com



Figura 5.4: Apartat d'administració del lloc web



Figura 5.5: Taules de la base de dades

Base de dades db taula productes

Registre nou

Producte

Marca

Model

Color

Descripció

Figura 5.6: Formulari per a afegir un nou producte a la web.

Base de dades db taula productes Id de registre
47

Editar el registre actual

Id 47

Producte

Marca

Model

Figura 5.7: Formulari per modificar un producte existent a la web.

6 Configuració dels mòduls del sistema

6.1 Introducció

En aquesta secció s'explicaran els passos seguits per tal de configurar i aconseguir l'òptim funcionament dels diferents mòduls i la configuració addicional necessària per poder aconseguir un òptim i complet coneixement del mòdul.

6.2 Mòdul de temperatura

El primer dels sensors que es va provar va ser el de temperatura, a priori es presentava com el més fàcil, ja que durant el grau de sistemes TIC s'havia treballat amb sensors de temperatura, concretament amb l'[arduino](#). Es pretenia estudiar el hardware necessari per al sensor, fer les proves més bàsiques per tal de verificar-ne el funcionament i a partir d'aquí fer un programa que ens anés dient la temperatura constantment.

A l'hora d'escollir el sensor, senzillament va ser necessari informar-te per internet dels diferents tipus de sensors de temperatura per utilitzar amb una [raspberry](#), i no va ser necessària molta estona de recerca per veure que tots recomanaven utilitzar el sensor de temperatura DS18B20. Per tal d'estudiar el sensor i veure el muntatge necessari es pot consultar el seu datasheet. [16]. Per configurar i obtenir les dades d'aquest s'ha seguit un tutorial, verificat prèviament amb fonts d'informació. [14]

Abans de començar a explicar les configuracions de hardware, se citen els components que formaran el mòdul. En el mòdul de temperatura, com bé diu el nom els components que en formaran part seran:

- Alimentador micro USB 5V
- Raspberry Pi 3 model B
- Cable coure rígid
- Sensor temperatura (DS18B20)
- Resistència de $4.7k\Omega$

6.2.1 Hardware

Podem veure l'esquema de pins del sensor DS18B20 a la figura 6.1.

Recordem també la disposició de pins del GPIO de la raspberry de l'apartat 4.2.1, per tal de poder entendre les connexions que es mostren a la taula 6.1 i la figura 6.2.

Una vegada coneixem les connexions necessàries, cal conèixer també que entre la pota de +VCC i la de DADES haurem de connectar una resistència de $4.7k\Omega$. [Figura 6.3]

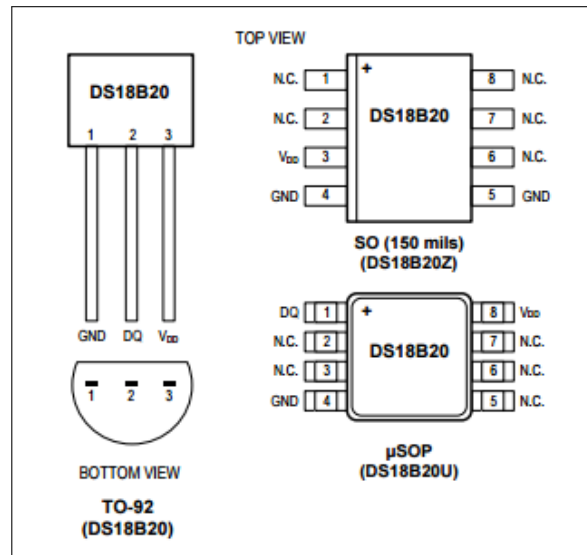


Figura 6.1: Pins del sensor DS18B20

DS18B20	GPIO
Pin 1 (GND)	Pin 3 (GND)
Pota 2 (Dades)	Pin 7 (GPIO 4)
Pota 3 (VCC)	Pin 1 (VCC 3.3V)

Taula 6.1: Taula de connexions del sensor DS18B20 i el GPIO de la raspberry.

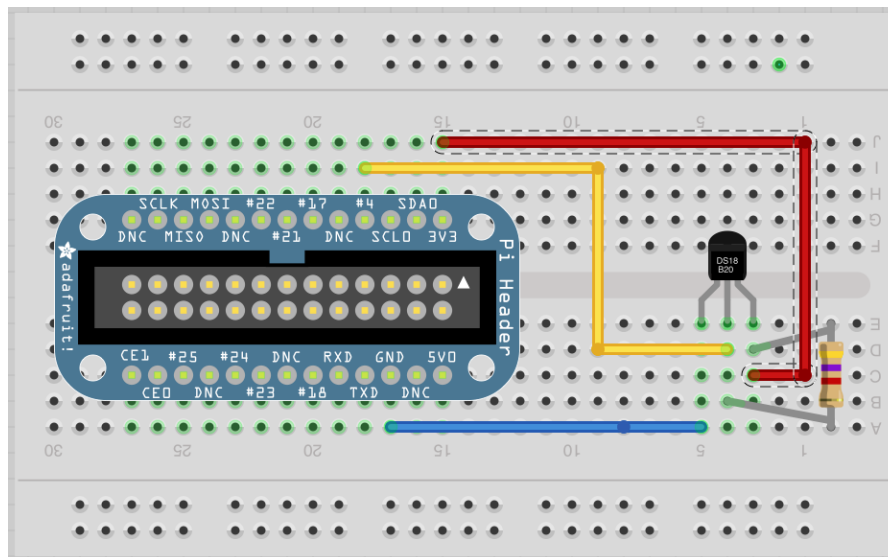


Figura 6.2: Connexions DS18B20 amb la raspberry.

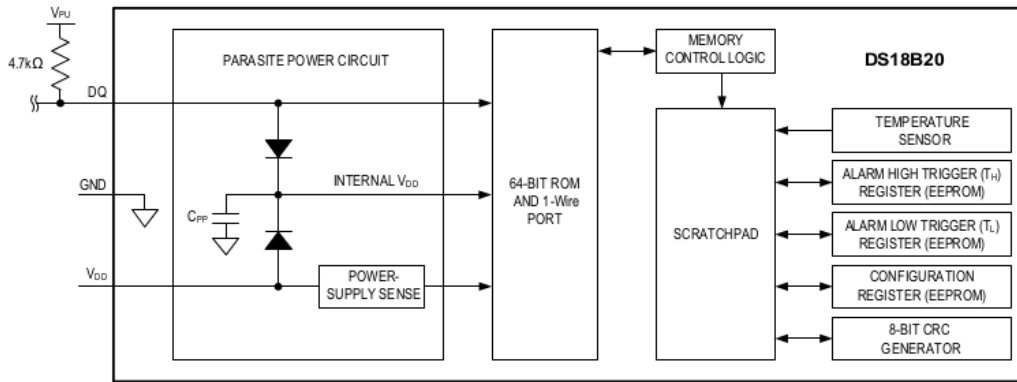


Figura 6.3: Esquema connexions DS18B20, i resistor 4.7KΩ

6.2.2 Software

Una vegada tenim el sensor connectat correctament a nivell hardware, queda només configurar a nivell software per tal de poder obtenir la temperatura.

Inicialment hem de modificar el fitxer `/boot/config.txt` afegint la següent línia al final d'aquest, una vegada afegida, caldrà reiniciar. Aquesta línia serveix per configurar el mòdul d'interfícies [1-wire](#).

```
dtoverlay=w1-gpio
```

A continuació, caldrà afegir els mòduls necessaris per al sensor al [kernel](#) de Linux. `w1-gpio`, `w1-therm`

```
sudo modprobe w1-gpio
sudo modprobe w1-therm
```

Finalment només ens caldrà accedir al directori `/sys/bus/w1/devices/` on trobarem un directori que començarà per `28-xxx` i un seguit de números, accedim a aquest directori i ja podem consular el fitxer que ens donarà la temperatura actual. [Figura 6.4]

```
cat w1_slave
```

```
pi@raspberrypi ~ $ cat /sys/bus/w1/devices/28-0000079f9eee/w1_slave
ba 01 4b 46 7f ff 06 10 d0 : crc=d0 YES
ba 01 4b 46 7f ff 06 10 d0 t=27625
```

Figura 6.4: Lectura del fitxer `w1_slave`, el qual ens mostra la temperatura actual 27.625

A continuació, en el codi 6.5, podem veure el codi que llegir l'arxiu `w1_slave` i posava la temperatura d'una forma més entenedora, i no amb els números que hem vist a la figura 6.4. I en la figura 6.6 podem veure l'execució d'aquest codi.

6.2.3 Problemes

Es vol citar també els problemes que s'han trobat a l'hora de tractar amb els sensors de temperatura. Inicialment es van comprar 5 sensors DS18B20, i a l'hora de provar els sensors, es

```
import os
import glob
import time

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c, temp_f

while True:
    print(read_temp())
    time.sleep(1)
```

Figura 6.5: Codi que ens retorna constantment la temperatura actual, Font: [14]

```
pi@raspberrypi ~ $ python thermometer.py
(27.687, 81.8366)
(27.625, 81.725)
(27.687, 81.8366)
(27.687, 81.8366)
(27.687, 81.8366)
```

Figura 6.6: Execució del fitxer `thermometer.py` amb el codi 6.5.

va consultar el [datasheet](#) per tal de conèixer les connexions necessàries per al correcte funcionament del sensor. Tot i així el primer sensor que es va provar es va sobreescalfar molt i per tant es va malmetre. Vistes les conseqüències es va tornar a provar amb un nou sensor, i tot i connectar-ho de la mateixa forma, semblava que tot i seguir els passos explicats anteriorment, el directori que comença amb `28-xxx` dins de `/sys/bus/w1/devices/` no apareixia.

Vist l'èxit i amb l'ajuda del tutor es va procedir a realitzar les proves amb l'oscil·loscopi del laboratori i controlant quina era la sortida del sensor en tot moment. En el moment de donar la tensió de 3.3V a un nou sensor, va tornar a cremar-se. A continuació es va decidir tornar a fer proves amb els sensors espatllats, i veiem que la sortida era de l'ordre de 1V, quan hauria de ser de 3.3V. Tot i això semblava que enviaven els bits corresponents a la temperatura. Es van provar els altres sensors, i si que semblava que enviaven les dades, però finalment i després de diverses proves, com per exemple posant 0V a la pota de dades, es va veure que no feia el pols de baixada correctament i que per tant els sensors adquirits eren defectuosos. Al detectar això es van comprar uns altres sensors de temperatura, i simplement fent el procés explicat en el punt anterior el directori `28-xxx` va aparèixer i consultant el fitxer `w1_slave` es va veure la temperatura.

S'han explicat aquests problemes, ja que inicialment es va dedicar moltes hores a fer funcionar el sensor, ja que el primer sensor adquirit era defectuós. Tot i això vam aprendre com comprovar que un sensor era defectuós, i quins eren els senyals i les alteracions que patia si es connectava de forma errònia.

6.2.4 Conclusions mòdul

Fins ara hem parlat només del sensor de temperatura i en aquest apartat ens referim al mòdul que incorpora el sensor de temperatura. Amb les connexions explicades i la configuració de la [raspberry](#) adequada, ja tenim implementat i configurat el mòdul de temperatura. Ara es pot muntar en qualsevol indret, sempre que disposem de connexió a internet i de corrent, només seguint els anteriors passos.

Aquest mòdul, pot acabar sent més complex, ja que es pot incorporar gràfiques i estadístiques per tal que la informació que se li pugui demanar sigui la màxima possible.

6.3 Mòdul reflector

El sensor CY-191B-P-Y, sensor reflector escollit, requereix la utilització d'un microcontrolador. Com ja s'ha dit en l'apartat dels mòduls es van mirar diversos sensors abans d'escollir aquest.

Una vegada escollit però i veient la informació i la fàcil configuració que requereix, s'ha vist que l'opció escollida era una de les millors opcions.

En el datasheet del sensor ens parla també de com hem d'utilitzar-lo si ho volem fer en [parasite mode](#), i dóna una explicació molt precisa de quina ha de ser la configuració per a aconseguir un correcte funcionament. [4]

S'adjunta plànol de l'entrada de l'establiment, per tal de facilitar la comprensió a l'hora de la col·locació dels 3 sensors reflectors utilitzats en el projecte. [Figura 6.7]

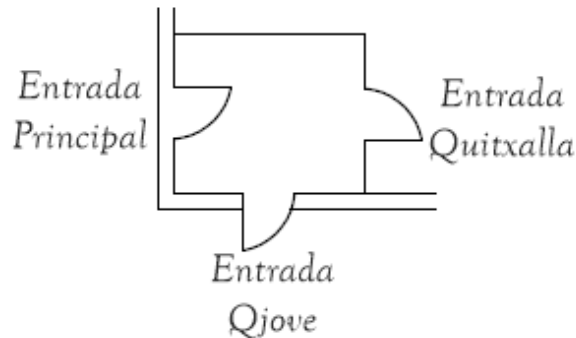


Figura 6.7: Plànol de l'entrada i ubicació dels sensors reflectors

6.3.1 Hardware

El hardware d'aquest sensor consistirà a connectar adequadament els cables de sortida del sensor [Figura 6.8] al GPIO i a l'alimentació corresponent. Es pot veure de forma més ordenada a la taula 6.2.

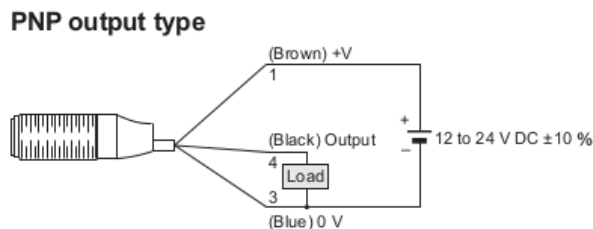


Figura 6.8: Connexions potes sensor CY-191B-P-Y.

CY-191B-P-Y	GPIO	Voltatge (V)
Cable Marró (GND)	-	GND
Cable Negre (Dades)	Pin 11 (GPIO 17)	-
Cable Blau (VCC)	-	+12V

Taula 6.2: Taula de connexions del sensor DS18B20 i el GPIO de la raspberry.

Cal saber també que els tres sensors instal·lats, al estar els tres a l'entrada de la botiga, estan connectats a la mateixa [raspberry](#). A la taula 6.3 es veuen les connexions dels tres sensors a

la raspberry.

S'adjunta la relació de sensors amb els pins del GPIO per si al llarg del treball es parla d'algun dels pins utilitzats, que aquesta taula ens pugui ser de gran ajuda.

Sensor	Localització	Pin GPIO
1	Porta Quitxalla	Pin 11 (GPIO 17)
2	Porta Qjove	Pin 13 (GPIO 27)
3	Porta exterior	Pin 15 (GPIO 22)

Taula 6.3: Taula de la ubicació dels sensors a la porta corresponent.

6.3.2 Software

Pel que fa al software dels sensors reflectors, en el present projecte es vol aconseguir tenir un control a temps real de la quantitat de persones que estan a dins l'establiment. Per aconseguir-ho s'ha implementat un programa amb `python reflective.py`, el qual s'ha programat utilitzant l'estructura d'un autòmat, podem veure la màquina d'estats que controla els sensors reflectors a la figura 6.9.

Per entendre la màquina d'estats cal recordar el següent:

- Porta Quitxalla = GPIO17
- Porta Qjove = GPIO27
- Porta exterior = GPIO22

Aquesta màquina d'estats, així com la implementació del codi, té diferents casos en els quals no acaba de ser del tot robust. Se n'expliquen alguns exemples a continuació.

1. Una persona passa pel sensor del carrer (GPIO22), a continuació, i abans que la primera superi un dels dos interns (GPIO17 o GPIO27), una altra torna a passar per primer sensor. A continuació una de les dos persones presents en l'entrada entra a una de les dues botigues. L'altra, torna a sortir per la porta del carrer. Aquest cas fallaria, ja que la màquina d'estats detectaria com si estiguéssim en l'estat (2+ GPIO22) quan en realitat estem en estat (Waiting).
2. També ens podem trobar el cas que en el primer sensor (GPIO 22) passin dues persones juntes, i que després se separin. En aquest cas la màquina es quedaria en l'estat (After GPIO17 o After GPIO27).

Hi ha decisions que s'han hagut de prendre pensant amb la probabilitat. Per exemple, dues persones han passat per la porta de fora i es detecta una tercera vegada el sensor de la porta de fora. En aquest cas podria ser que una de les dues persones hagi marxat, o bé que hagués entrat una tercera. En aquesta ocasió s'ha decidit pensar que hi haurà més vegades que entrarà una tercera persona que no pas en sortirà una de les dues de dins.

Tot i aquests problemes, abans de dissenyar i programar el programa que controla això ja se n'era conscient. Tot i així, coneixent que si posàvem dos sensors, un al costat de l'altre, a la

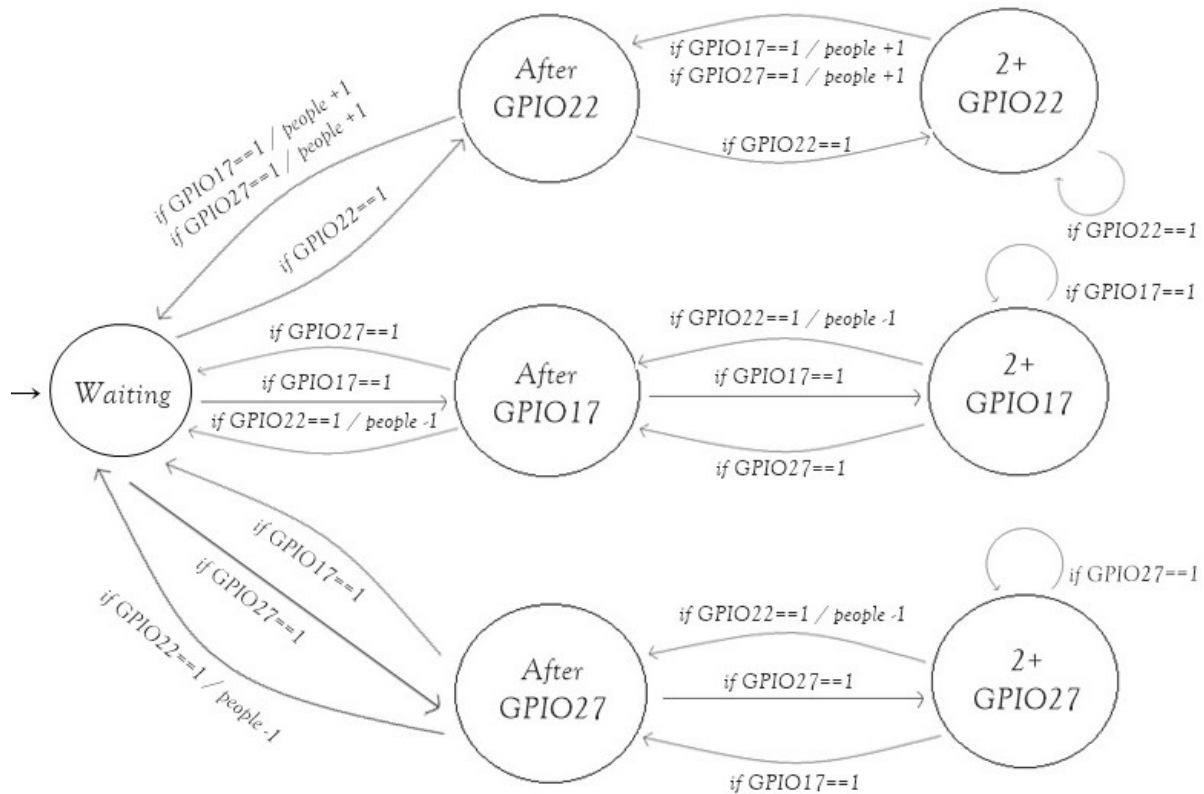


Figura 6.9: Màquina d'estats corresponent a la gestió dels sensors reflectors (reflective.py).

porta exterior, podríem tenir en tot moment i sense tanta complicació el nombre de persones que hi ha dins l'establiment, es va optar per una solució software. Es va optar però per a una solució software, entenent que la solució hardware, sempre es pot fer, i no ha de provocar cap complicació. La solució d'aquest problema en àmbit de software es va trobar força atractiva, tot i la seva complexitat. D'aquesta forma es pot controlar a quina de les dues botigues entren i surten les persones. Si es volgués controlar això a nivell de hardware, caldria posar 2 sensors a cada porta, augmentant al doble el nombre de sensors necessaris, així com el cost del treball.

En el codi 6.1 podem veure una part del codi que correspon al control dels sensors reflectors. Es pot veure complet en els annexos.

Codi 6.1: Part del codi `reflective.py`

```

def state_machine(sensor):
    global state, people, entrance
    if GPIO.input(22): #Carrer
        if state == 0:
            state = 1
        elif state == 1:
            entrance += 1
            state = 4
  
```

```

    elif (state == 2) or (state == 3):
        state = 0
        people -= 1
    elif state == 4:
        entrance += 1
    elif state == 5:
        people -= 1
        entrance -= 1
        if entrance == 0:
            state = 0
    elif state == 6:
        people -= 1
        entrance -= 1
        if entrance == 0:
            state = 0
elif GPIO.input(17): #Quitxalla
    if state == 0:
        state = 2
    elif state == 1:
        state = 0
        people += 1
    elif state == 2:
        entrance += 1
        state = 5
    elif state == 3:
        state = 0
    elif state == 4:
        people += 1
        entrance -= 1
        if entrance == 0:
            state = 0
    elif state == 5:
        entrance += 1
    elif state == 6:
        entrance -= 1
        if entrance == 0:
            state = 0
elif GPIO.input(27): #Qjove
    if state == 0:
        state = 3
    elif state == 1:
        state = 0
        people += 1
    elif state == 2:
        state = 0
    elif state == 3:

```

```
        entrance += 1
        state = 6
    elif state == 4:
        people += 1
        entrance -= 1
    if entrance == 0:
        state = 0
    elif state == 5:
        entrance -= 1
        if entrance == 0:
            state = 0
    elif state == 6:
        entrance += 1
```

```
GPIO.add_event_detect(17, GPIO.RISING, callback=state_machine)
GPIO.add_event_detect(27, GPIO.RISING, callback=state_machine)
GPIO.add_event_detect(22, GPIO.RISING, callback=state_machine)
```

6.3.3 Problemes

Pel que fa al mòdul reflector no hi van haver problemes destacats a niell de hardware, ja que al tenir un bon [datasheet](#) i al tractar-se d'uns sensors *Panasonic* a priori d'una bona marca, la configuració i instal·lació va funcionar correctament.

Cal dir per això, que perquè funcionin els sensors reflectors s'ha d'utilitzar un catadiòptric perquè reboti el raig emès. A part d'això no es pot utilitzar un simple mirall, ja que els raigs no es polaritzen, per tant funcionen únicament amb un catadiòptric.

Per últim, dir que els problemes que ens vam trobar pel que fa a software, han estat explicats en el punt anterior, aprofitant l'explicació de la màquina d'estats.

6.3.4 Conclusions mòdul

El mòdul reflector, format pel sensor reflector i la [raspberry](#), serà un altre mòdul intel·ligent, el qual estarà constantment esperant rebre senyals del sensor. Cal saber també que per poder detectar la direcció d'un objecte o persona serà necessari disposar de dos mòduls reflectors, situant els sensors l'un al costat de l'altre. També podem evitar l'ús d'una raspberry, ja que el [GPIO](#) permet la connexió de múltiples sensors.

En el present projecte s'han ajuntat 3 mòduls reflectors, connectats però, a la mateixa raspberry, amb la finalitat de detectar la direcció de les persones, i també a quina de les dues botiga entren o surten.

Aquest mòdul, i una vegada configurat per detectar quan es talla el raig del sensor, podrem executar-hi els programes desitjats, ja sigui per tal de fer estadístiques o controls de gent, o bé per detectar el pas de productes.

En el cas que ens ocupa, es contarà i per tant es tindrà control de la quantitat de clients que hi ha a temps real dins de l'establiment.

Com bé sabem, cada mòdul és altament configurable i versàtil a l'hora del tractament de la informació i per tal de l'obtenció de les estadístiques.

6.4 Mòdul càmera

Aquest mòdul és el més senzill, ja que l'única complexitat que requereix, és la configuració de la càmera. Ens servirà per tenir a temps real i des de qualsevol punt, una visió de la botiga, i per tant utilitzar aquest mòdul com a un element més de seguretat.

6.4.1 Hardware

Aquest mòdul, pel que fa a hardware, consta només de connectar la càmera IP a la corrent amb un adaptador de 5V, i a internet mitjançant el cable de xarxa.

6.4.2 Software

A nivell de nivell de software, es podia escollir dos camins, configurar-ho mitjançant internet, o bé amb l'aplicació mòbil. Si es feia via internet, les opcions a configurar eren moltes més, però per contra, havia de ser instal·lant un connector específic, amb el sistema operatiu Windows, i amb el navegador Internet Explorer [Figura 6.10]. En canvi amb l'aplicació [Figura 6.11], tot i tenir menys opcions van esser suficients per aconseguir poder rebre el vídeo amb [RTSP](#) a partir de la IP assignada.

Per a fer-ho va ser necessari crear un usuari a la càmera i així poder-hi accedir només si se'n coneixia l'accés.



Figura 6.10: Captura del navegador a l'hora de configurar la càmera Foscam C1

6.5 Mòdul servidor raspberry

6.5.1 Preparació inicial de la raspberry

Al disposar només d'una raspberry, dues més eren necessàries. Per això es va comprar la raspberry pi 3, per si es volia fer la comunicació amb wifi, tot i finalment no utilitzar-ho. Després de veure la comparativa [23], es va comprar per amazon. [1]

Primer de tot calia configurar la raspberry per tal de que poguéssim connectar-nos-hi des del ordinador i així poder fer totes les proves necessàries i configurar els sensors.

Després d'instal·lar rasbian, el sistema operatiu de la raspberry, es va configurar l'accés a internet a la raspberry. Es va donar una IP estàtica a cada una de les raspberry, (192.168.1.13,

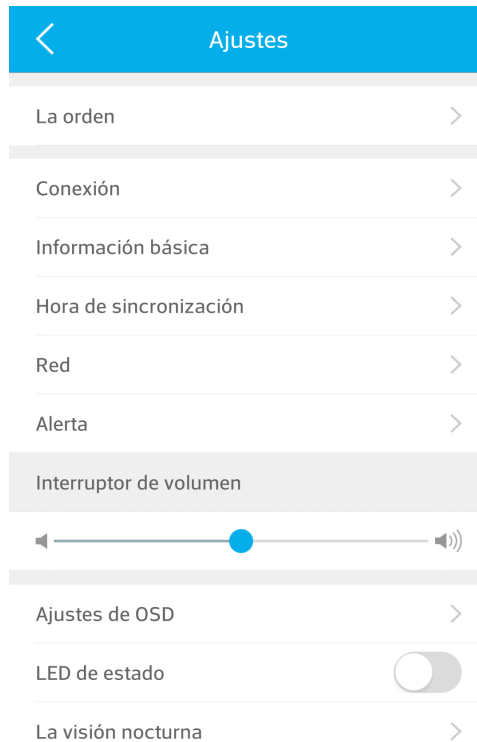


Figura 6.11: Captura de l'aplicació del mòbil a l'apartat de configuració

192.168.1.14 i 192.168.1.15). Després se li va dir quina IP té el router (gateway) i la màscara, tot això en el fitxer `/etc/network/interfaces`.

```
auto lo

iface lo inet loopback
iface eth0 inet static
    address 192.168.1.14
    netmask 255.255.255.0
    gateway 192.168.1.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Inicialment es va connectar diverses vegades la raspberry al ordinador, per tal de fer proves. Per configurar el internet s'ha de canviar el *gateway* i posar-hi la IP del nostre ordinador, ja que farà de [router](#).

Quant connectem la raspberry al router, aquest fa [NAT](#) automàticament als paquets, a diferència del ordinador, el qual hem d'activar el [forwarding](#) i afegir *wlan0* a la taula NAT.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
```

Finalment hem d'afegir el rang de la nostra raspberry a la interfície *eth0*.

```
sudo ifconfig eth0 192.168.1.20
```

Amb els passos descrits anteriorment, hem de ser capaços de configurar l'accés a internet de la nostra raspberry.

6.5.2 Servidor apache per servir web2py

Una vegada es va haver dissenyat tota la web, s'havia de servir els fitxers mitjançant un servidor web. Per fer-ho es va utilitzar [apache](#). Web2py incorpora un apartat en la seva documentació que t'ajuda a crear el fitxer de configuració de apache, per tal de servir l'aplicació web2py. Podem veure en el codi 6.2 quina va ser finalment la configuració SSL, per tal de servir el contingut web.

Codi 6.2: Arxiu de configuració d'apache

```
<VirtualHost *:443>
    ServerName laquitxalla.ddns.net
    Redirect "/welcome/default/index" "/quitxalla/default/index"
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/2_laquitxalla.com.crt
    SSLCertificateKeyFile /etc/apache2/ssl/laquitxalla.key
    SSLCertificateChainFile /etc/apache2/ssl/1_root_bundle.crt

    WSGIProcessGroup web2py

    WSGIScriptAlias / /home/www-data/web2py/wsgihandler.py

    <Directory /home/www-data/web2py>
        AllowOverride None
        Order Allow,Deny
        Deny from all
        <Files wsgihandler.py>
            Allow from all
        </Files>
    </Directory>

    AliasMatch ^/([^/]+)/static/(?:_[\d]+\.[\d]+\.[\d]+)/?(.*) /
        home/www-data/web2py/applications/$1/static/$2

    <Directory /home/www-data/web2py/applications/*/static/>
        Order Allow,Deny
        Allow from all
    </Directory>
```

```
CustomLog /var/log/apache2/access.log common
ErrorLog /var/log/apache2/error.log

</VirtualHost>
```

Ens els annexos es poden veure un seguit d'imatges corresponents a les plaques de circuit imprès i a les connexions dels cables corresponents a les connexions entre sensors i raspberry. Finalment cal citar també que s'han emprat les aplicacions de *ddns*, *noip* i *startssl*, per tal de fer que la pàgina web tingui un certificat, i que s'actualitzi la IP dinàmica del router automàticament. Tot i la creació d'un certificat gratuït per a www.laquitxalla.com, al utilitzar un servei gratuït (*noip*), ens redirigeix al domini <https://laquitxalla.ddns.net> i al no ser-ne el propietari, no s'ha pogut crear un certificat per aquest domini.

7 Implementació del sistema final

7.1 Introducció

Tal com hem vist, en el present projecte s'han creat mòduls amb una gran versatilitat per tal de poder crear un sistema final al gust del consumidor, i així poder modificar i ampliar el sistema que s'explica a continuació.

Per a la construcció del nostre sistema final utilitzarem els següents mòduls:

- 1 mòdul càmera IP
- 1 mòdul servidor
- 3 mòduls de temperatura
- 3 mòduls reflectors

S'han escollit els anteriors mòduls, per disposar primerament, d'una càmera de seguretat que controlarà quasi bé tota la botiga. El servidor, essencial per a poder gestionar tot el tema de la pàgina web, així com l'obtenció i gestió de les dades que ens serviran els altres mòduls.

La botiga consta d'uns $170m^2$ per tant la temperatura en diferents indrets de la botiga pot ser lleugerament diferent. Per aquest fet s'ha decidit muntar 3 mòduls de temperatura, un a cada extrem del local, i el darrer estarà ubicat a una columna just al costat del taulell, ja que és on més ubiquen els clients. Finalment, els 3 mòduls reflectors, i tenint constància dels entrebancs i solucions plantejats en l'apartat 6.3.3, s'han instal·lat un a cada una de les 3 portes que formen l'entrada de l'establiment.

Per tenir una idea inicial de la ubicació de cada element dins de l'establiment, s'adjunta un plànol de la botiga amb la informació referent a la majoria dels elements utilitzats en el projecte, i per tant en el sistema final. [Figura 7.1]

A continuació s'explicarà on i com es va muntar cada mòdul a la botiga, i es podrà veure instantànies una vegada finalitzada la instal·lació.

7.1.1 Instal·lació mòduls de temperatura

Sensor 1 (Fons de la botiga)

Per a la instal·lació del primer sensor, ubicat al fons de la botiga, va ser necessari fer-hi arribar el corrent elèctric per a poder endollar l'adaptador de 5V per la [raspberry](#), tot i això va ser relativament senzill, ja que vam poder agafar el corrent del focus més proper. Tot el cablejat es va passar per sobre les plaques de la botiga, també s'hi han deixat les raspberry i les plaques de circuit imprès realitzades.

Una vegada passat el cable de corrent elèctric fins al punt desitjat, així com el cable de xarxa provinent del commutador de xarxa, ubicat al centre de la botiga, simplement restava foradar

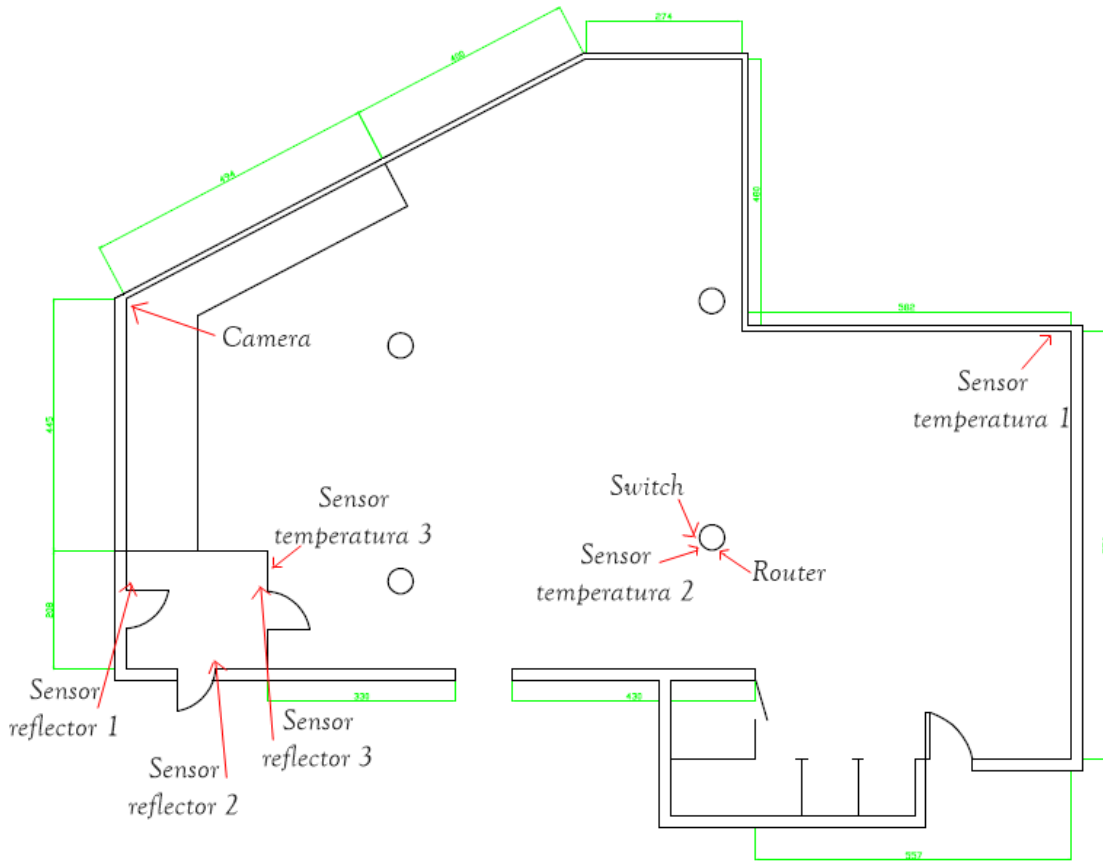


Figura 7.1: Ubicació dels diferents components del sistema final en la botiga.

la placa i fer sortir el sensor reflectiu com veiem en la figura 7.2. En la imatge, també es pot veure la ubicació del sensor de temperatura situat al fons de la botiga.

Sensor 2 (Centre de la botiga)

Per a la instal·lació del sensor 2, situat en el centre de la botiga va ser necessari baixar els cables a través de la columna, els quals es connectarien a la raspberry situada al mateix punt que el sensor 2, però a sobre les plaques. El sensor número 2 es va ubicar a l'alçada dels clients, a sobre el regulador de temperatura de la botiga. En la figura 7.3 podem veure la ubicació, i també la columna per la qual vam haver de baixar els cables.

Sensor 3 (Entrada de la botiga)

L'últim sensor de temperatura es va instal·lar a sobre la porta d'entrada a la botiga, corresponent també a l'extrem contrari al sensor 1. Es va aprofitar un espai que hi ha sobre l'entrada de la botiga per la col·locació de la raspberry i de la placa de circuit imprès, i una càmera que no funciona que hi ha just a sobre la porta d'entrada a la Quitxalla. [Figura 7.4].



Figura 7.2: Sensor de temperatura 1: Ubicació i visió final d'aquest.



Figura 7.3: Sensor de temperatura 2: Ubicació i visió final d'aquest.

7.1.2 Instal·lació mòduls reflectors

Just sobre l'entrada de la botiga hi ha un espai el qual ha estat utilitzat per a connectar tots els cables dels sensors reflectors i la raspberry en la qual han estat connectats. Va ser necessari foradar la fusta, sostre de l'entrada, per poder passar els cables de baix cap a dalt. Els sensors



Figura 7.4: Sensor de temperatura 3: Ubicació i visió final d'aquest.

reflectors, van ser col·locats a una alçada de 70 cm del terra, perquè més avall detectarien les dues cames d'una persona com a dues persones, i més amunt els nens petits no serien captats. A continuació s'explica la informació específica a l'hora d'instal·lar cada sensor.

Sensor reflector 1 (Entrada principal)

Pel que fa al primer sensor reflector, ubicat a la porta principal de la botiga, cal dir que és el sensor que treballa a més distància, concretament a 1.5 m. Aquesta porta és doble, i degut al disseny d'aquesta l'única solució possible va ser instal·lar el sensor tal com veiem en la figura 7.5. En aquesta imatge, també podem veure el cable que puja perpendicular a terra, el qual acaba pujant a la part superior, per a ser connectat a la raspberry.



Figura 7.5: Sensor reflector 1: Ubicació i visió final d'aquest.

Sensor reflector 2 (Entrada QJove)

El segon sensor, es va instal·lar a la porta d'entrada a la Q Jove. Aquest es va col·locar al marc de la porta, i el catadiòptic a la paret del davant. Es van tallar els catadiòptrics, per tal de deixar només la part reflectora, i no les pestanyes negres pensades per a la seva fixació. Els catadiòptrics van ser enganxats amb pega a una paret blanca, i a l'haver tallat les pestanyes que sobresortien dels extrems, va quedar molt més dissimulat i elegant. [Figura 7.6]



Figura 7.6: Sensor de reflector 2: Ubicació i visió final d'aquest.

Sensor reflector 3 (Entrada Quitxalla)

L'últim dels sensors va ser situat a l'entrada de la Quitxalla, part de la botiga en la qual s'ha realitzat el projecte. Per al muntatge d'aquest sensor, col·locat en el marc esquerre de la porta, va ser necessari aplicar-li una lleugera inclinació, per tal de què el raig de llum, rebotés a l'interior del marc dret de la mateixa porta. Es va fer així perquè així evitàvem que el raig del sensor 2 i 3 es creuessin en la cantonada de l'entrada. [Figura 7.7]



Figura 7.7: Sensor reflector 3: Ubicació i visió final d'aquest.

7.1.3 Instal·lació mòdul càmera

El mòdul càmera format simplement per la càmera, va quedar ubicat en la cantonada esquerra de la botiga (veure plànol 7.1). En la figura 7.8 es pot veure la instal·lació i ubicació final d'aquest mòdul.



Figura 7.8: Càmera: Ubicació i visió final d'aquesta.

7.1.4 Instal·lació mòdul servidor

Finalment el mòdul servidor, va ser ubicat sobre les plaques, en el centre de la botiga, i a vora el commutador de xarxa. En el plànol 7.1 es pot veure on hi ha l'encaminador i el commutador, mateixa ubicació de la raspberry servidor. [Figura 7.9]

7.2 Sistema final

En la figura 7.10 podem veure com ha quedat el sistema final a nivell de hardware per tal de tenir una visió completa de les connexions entre els mòduls utilitzats.

Podem veure el mòdul de temperatura dins del rectangle **vermell**, el mòdul reflector en el **verd**, el mòdul càmera IP dins del **groc** i finalment el servidor en el **blau**.

7.3 LOPD - Llei Orgànica de Protecció de Dades

Arribat en aquest punt, és necessari explicar que en el present projecte es tracta amb temes legals, pel fet que hi ha una càmera de seguretat enregistrant als clients en tot moment. Per això, i com que el projecte realitzat és un projecte real, i totalment operatiu des de fa unes setmanes, s'ha cregut necessari explicar que s'ha hagut de fer per tal de tenir els temes legals en ordre.

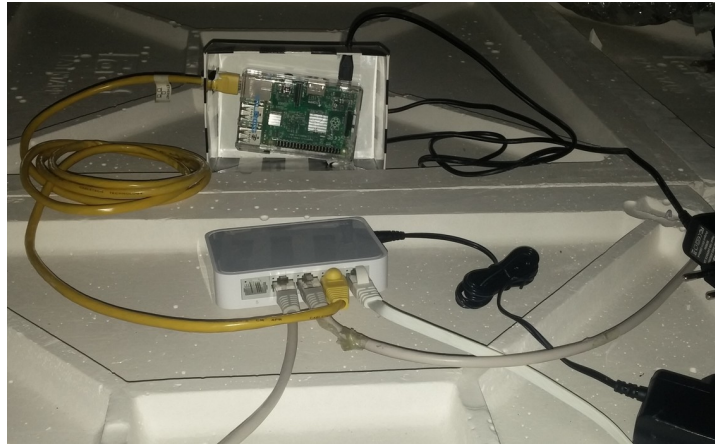


Figura 7.9: Raspberry servidor: Ubicació i visió final d'aquesta.

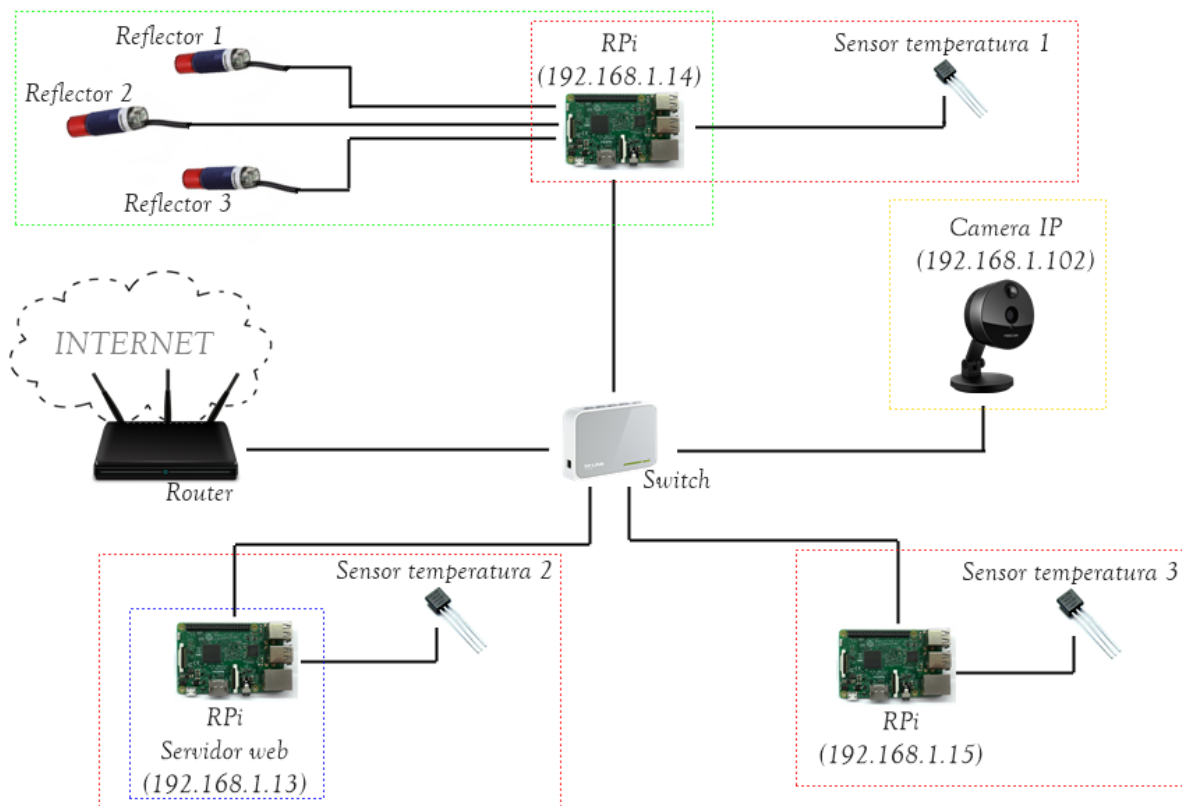


Figura 7.10: Interconnexions de la infraestructura completa del projecte.

Inicialment es va mirar com haviem de fer per registrar-nos a la LOPD. A continuació es va veure que hi havia empreses que s'hi dedicaven, facilitant-te la documentació necessària així com una formació per als responsables. Després de buscar diverses empreses es va trobar una que ens oferia donar-nos d'alta a la LOPD i la gestió dels papers per 120 €+ IVA. Es va escollir aquesta opció.

L'empresa ens va demanar quin tipus de dades serien les que nosaltres enregistraríem, i per el present treball, només hi haurà la càmera de seguretat, a partir de la qual es podrà veure i enregistrar vídeos o imatges en qualsevol moment del dia. Més endavant però, es voldrà vendre per internet, així que s'haurà de modificar alguna característica del registre.

Una vegada explicat això l'empresa ens va fer arribar diversos formularis, els quals vam anar omplint i reenviant, per tal que es pogués tramitar la sol·licitud de registre a la LOPD.

Arribat a aquest punt i al ser un tema lent i en el qual s'ha de tenir especial vigilància, a dia d'avui la sol·licitud no ha estat acceptada encara, ja que el procés requereix unes quantes setmanes. Tot i així es vol deixar constància que està tot omplert i sol·licitat, i que estem a l'espera de l'últim pas, l'aprovació de l'*Agència Espanyola de Protecció de Dades*.

Una vegada aprovada la sol·licitud, restarà només afegir la placa per advertir que en aquell establiment seran enregistrats.

8 Integració del sistema final a la web

Una vegada hem vist els mòduls que es disposen, i la configuració i implementació d'aquests, només queda la integració final al lloc web. Per a la integració, serà necessari obtenir les dades de tots els mòduls, i a continuació tractar-les, si és precís, i incorporar-les a l'apartat d'administració de la pàgina web.

Finalment el sistema ha quedat repartit amb la [raspberrry](#) servidor, al centre, dues raspberrry més una a cada extrem de la botiga, i la càmera, a l'extrem esquerre. Per tant les dues raspberrry dels extrems i la càmera han estat connectades mitjançant el cable de xarxa al commutador de xarxa.

Una vegada connectat a internet, podem passar a explicar com s'han obtingut les dades de cada mòdul, i com s'han integrat a la web.

8.1 Integració del mòdul temperatura

De mòduls de temperatura en tenim 3, i començarem explicant la integració del sensor número 2, ja que està a la mateixa raspberrry que el servidor web, i per tant no serà necessari la comunicació a través de la xarxa.

8.1.1 Sensor DS18B20 2

Aquest sensor, serà tractat de forma diferent, ja que està a la mateixa raspberrry, per tant els passos seguits per a l'obtenció de les dades han estat:

- **Creació del programa que llegeix la temperatura de forma periòdica:** Aquest programa és l'adjuntat en l'apartat de configuració 6.2.2.
- **Modificació del programa anterior:** A partir de l'anterior programa, es van fer lleugeres modificacions per tal què el programa retornés només una lectura de la temperatura actual, i no la temperatura cada segon. [Codi 8.1]
- **Integració del programa en el controlador de la pàgina web:** A continuació es va integrar el codi en la funció *temp()* del controlador `default.py`, juntament amb el codi dels altres dos sensors que s'explicaran a continuació.
- **Creació del fitxer HTML `temp.html`:** En el fitxer [HTML](#), serà necessari crear una divisió amb un identificador que serà el que utilitzarà el [script](#) per tal d'actualitzar el text d'aquesta divisió. El script 8.2 s'encarregarà de cridar a la funció del script 8.3 cada minut, que serà la funció que consultarà la temperatura actual mirant l'arxiu `temp.dat`.

```
import os
import glob
import time
os.system('modprobe w1-gpio')
os.system('modprobe w1_therm')
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_c = str(temp_c) + " C"
    return temp_c

f = open('/home/www-data/web2py/applications/quitxalla/views/default/temp.dat', 'w')
f.write(read_temp())
f.close()
```

Figura 8.1: Part de la funció *temp()* del controlador `default.py`

```

<script>
  function temperature() {
    var xmlhttp;
    var objecte_div;
    objecte_div=document.getElementById("sensor2");
    xmlhttp=new XMLHttpRequest();
    xmlhttp.onreadystatechange=function() {
      if (xmlhttp.readyState==4 /*&& xmlhttp.status==200*/) {
        objecte_div.innerHTML=xmlhttp.responseText;
      }
    }
    xmlhttp.open("GET","temp.dat",true);
    xmlhttp.send();
  }
</script>

```

Figura 8.2: Script encarregat de consultar la nova dada del fitxer `temp.dat`

```

<script>
  <!--Cada 3 segons, crida la funcio temperature()-->
  setInterval("temperature()",60000);
</script>

```

Figura 8.3: Scripts encarregats de cridar la funció `temperature()` cada minut.

```

<div class="col-lg-4">
  <h5 class="thin"> Sensor (Centre botiga)</h5>
  <h3 class="thin" id="sensor2"> </h3>
</div>

```

Figura 8.4: Divisió HTML on s'identifica l'element `id="sensor2"`, el qual serà actualitzat amb javascript.

8.1.2 Sensor DS18B20 1 i 3

Els sensors 1 i 3, situats en el fons i porta de la botiga, estan connectats a un microcontrolador diferent. Per aquest fet, el mètode utilitzat ha estat completament diferent. Per aconseguir les dades d'aquests mòduls, s'ha fet el següent:

- **Creació script que inicia el servidor:** Aquest script escrit amb `bash`, executa una comanda la qual inicia un servidor `python` en el directori establert prèviament. [Codi 8.1]
- **Creació script que llegeixi la temperatura:** Aquest `script` és molt semblant al que hem vist en el codi 8.1, amb la lleugera diferència que deixa el fitxer `temp.dat`, que posteriorment llegirà el servidor web, al directori en el qual s'executa l'arxiu `server.sh`, que inicia el servidor.
- **Integració dels scripts al cron:** El `cron` ens permet als usuaris de sistemes Unix executar comandes o guions de `shell` de forma automàtica a una data i temps específics. El primer script ens servirà per iniciar el servidor, i per tant només cal que l'executem cada vegada que es reiniciï la raspberry. El segon script ha d'actualitzar la temperatura cada minut, per tal de seguir amb el mateix temps d'actualització del sensor número 2. Per editar el cron, i afegir les comandes que volem executar cada cert temps, podem executar la següent comanda:

```
sudo crontab -e
```

A continuació podrem editar aquest arxiu, i les línies que s'han afegit al final han estat les del codi 8.3.

- **Lectura de l'arxiu `temp.dat` del servidor:** Una vegada tenim els crons executant-se en les dues raspberry, la 192.168.1.14 (Entrada), 192.168.1.15 (Fons de la botiga), des del servidor web allotjat a la raspberry 192.168.1.13 (Centre de la botiga), i més concretament des de la web es farà `GET` dels arxius que contenen la temperatura. `temp1.dat` (Sensor 1, fons de la botiga), `temp3.dat` (Sensor 3, entrada de la botiga). Més concretament el `GET` és realitzat des del controlador `default.py`. [Codi 8.4]

Codi 8.1: Codi corresponent a l'arxiu `server.sh` present en les raspberry .14 i .15

```
cd /home/pi/tfg/
python -m SimpleHTTPServer
```

Codi 8.2: Part de la funció `temp()` del controlador `default.py`

```
# -*- coding: utf-8 -*-
import os
import glob
import time
os.system('modprobe_w1-gpio')
os.system('modprobe_w1-therm')
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
```



```

device_file = device_folder + '/w1_slave'
def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_c = str(temp_c) + "°C"
    return temp_c

f = open('/home/pi/tfg/temp1.dat', 'w')
f.write(read_temp())
f.close()

```

Codi 8.3: Línies de codi afegides al cron de la raspberry.

```

*/1 * * * * python /home/pi/tfg/temp.py
@reboot /home/pi/tfg/server.sh

```

Codi 8.4: Codi corresponent a la funció *temp()* del controlador *default.py*

```

#Sensor1
httpServ = urllib.HTTPConnection("192.168.1.15", 8000)
httpServ.connect()
httpServ.request('GET', "temp1.dat")
response = httpServ.getresponse()
temp1 = response.read()
httpServ.close()

#Sensor3
httpServ = urllib.HTTPConnection("192.168.1.14", 8000)
httpServ.connect()
httpServ.request('GET', "temp3.dat")
response = httpServ.getresponse()
temp3 = response.read()
httpServ.close()

```

Una vegada vist com s'ha incorporat la informació de tots els mòduls de temperatura, es pot veure com ha quedat la informació de temperatura representada en el lloc web en la figura 8.5.



Figura 8.5: Captura de pantalla de la informació de temperatura plasmada a la web.

8.2 Integració del mòdul reflector

Per a la integració dels mòduls reflectors a la web vam haver d'utilitzar un procediment molt similar al dels sensors de temperatura número 1 i 3. Se'n detallen els passos a continuació:

- **Creació script que inicia el servidor:** Aquest [script](#) és necessari perquè el mòdul servidor funcioni, però en el nostre cas, com que els sensors reflectors estan a la mateixa [raspberry](#) i utilitzarem el mateix directori per a la compartició d'arxius, no serà necessari tornar a iniciar el servidor.
- **Creació script que controla els sensors :** Aquest script és l'encarregat de controlar els sensors, i és el mateix que s'ha explicat a l'apartat del software del sensor. [Apartat 6.3.2]
- **Integració dels scripts al cron:** El programa anterior, s'haurà d'executar només una vegada, i aquest ja romandrà en l'estat de la base de dades corresponent. Per tant la comanda a incorporar al cron, serà la que veiem en el codi 8.5.

- **Lectura de l'arxiu `reflective.txt` del servidor:** Finalment i igual que en els sensors de temperatura, des del controlador del [framework](#) web2py, es farà el que veiem en el codi 8.6.

Codi 8.5: Línia afegida al cron de la raspberry.

```
@reboot /home/pi/tfg/reflective.py
```

Codi 8.6: Codi corresponent a la funció *people()* del controlador `default.py`

```
httpServ = httplib.HTTPConnection("192.168.1.14", 8000)
httpServ.connect()
httpServ.request('GET', "reflective.txt")
response = httpServ.getresponse()
people = response.read()
httpServ.close()
```

Ara és el moment de veure com queda en l'entorn web a l'hora de veure els clients actuals. [Figura 8.6]



Figura 8.6: Captura de pantalla de la informació corresponent als clients actuals.

8.3 Integració del mòdul càmera

La càmera, una vegada configurada, ens proporciona una IP, en la qual podem accedir i veure el vídeo en temps real, mitjançant el protocol [RTSP](#). El que es volia de la càmera, era integrar-la al lloc web, com si fos un vídeo, però en temps real.

Prova 1

Per tant, semblava que senzillament seria afegir una etiqueta al fitxer html desitjat, posant a la IP de la càmera com a font i amb els paràmetres d'usuari, contrasenya, port, i ruta. [Codi 8.7]

Codi 8.7: Codi inicial pensat per a la reproducció del vídeo en temps real.

```
<video width="600" controls>
<source src="rtsp://Quitxalla:camera-2016@laquitxalla.ddns.net:554/
videoMain>
</video>
```

Una vegada vist que no funcionava es va mirar quins eren els formats de vídeo que acceptaven els navegadors, i RTSP no és un protocol acceptat pels navegadors. De fet n'accepten molt pocs.

El que feia l'exemple anterior, era obrir un [pop-up](#) demanant si es volia executar l'ordre en una aplicació en local. Si es feia això, s'executava l'aplicació per defecte del nostre dispositiu, i si aquesta accepta el protocol RTSP, es podia veure el vídeo correctament. L'aplicació per defecte de vídeo d'Ubuntu, o bé el reproductor multimèdia VCL, són bons candidats i accepten el protocol RTSP.

Prova 2

Per tant la següent solució va ser incorporar el [plugin](#) de VLC en el codi HTML, per tal de què el navegador fos capaç d'executar la comanda i mostrar-nos el vídeo. Tampoc va funcionar.

Prova 3

Finalment, sabent que els navegadors accepten el format de vídeo HTTP, i que amb el VLC es podia transmetre el vídeo en aquest format, per tant vam intentar el que s'explica a continuació. En les figures 8.7, 8.8 i 8.9 veiem el procés a seguir per a transmetre el vídeo, agafant la font RTSP, establint HTTP i la ruta a transmetre, i finalment, el format escollit.

Una vegada seguits els passos anteriors, només restava accedir a la ruta especificada, per tal d'obtenir el vídeo en el protocol HTTP. Només funcionava si el flux s'executava des d'un altre VLC.

Prova 4

Seguint amb la mateixa idea, es va provar d'executar la comanda per transmetre el vídeo encapsulat en protocol HTTP a través del terminal.

```
cvlc rtsp://Quitxalla:camera-2016@laquitxalla.ddns.net:554/videoMain --sout '#
transcode{vcodec=theo,vb=400,width=368,height=208,acodec=none}:http{mux=ogg,
dst=:8082/video.ogg}'
```

Executant aquesta comanda i amb el codi 8.8, en l'ordinador es va aconseguir que el vídeo es reproduís en temps real, i sense obrir cap aplicació en local.

Codi 8.8: Codi final pensat per a la reproducció del vídeo en temps real.

```
<script>
document.vlc.playlistClear();
var options=[":audio-track=5"]; // select audio track 5 (=6th, 1st
is 0)
```

```

document.vlc.addTarget("...",options,2,0); // replace entry 0
document.vlc.play();
</script> —>

<video width="600" controls>
<source src="http://127.0.0.1:8082/video.ogv" type="video/ogg">
</video>

```

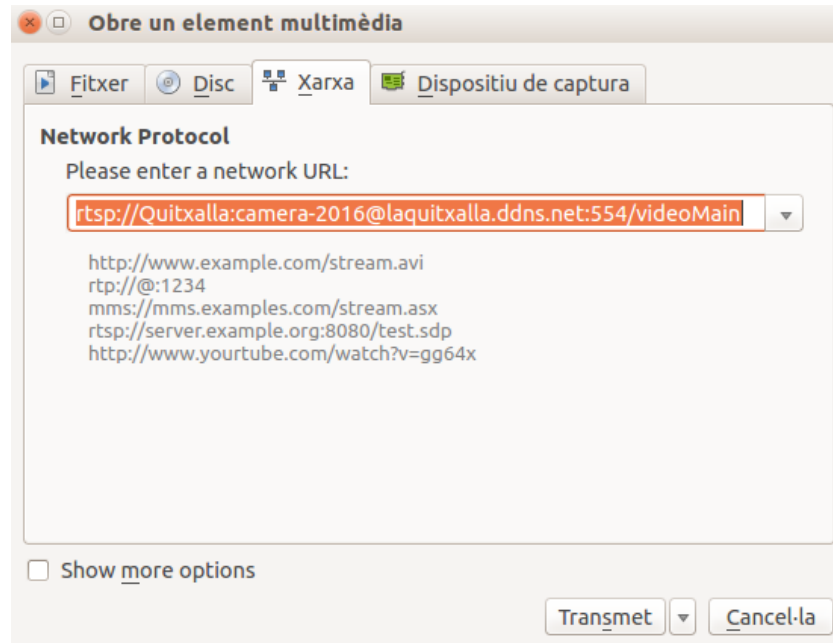


Figura 8.7: Captura del pas 1 de la transmissió del vídeo en protocol HTTP

Creient que ara sí que funcionaria la idea, a la que es va provar d'executar la comanda a la raspberry, no va funcionar. Donava diferents errors [Figura 8.10] que es van intentar solucionar, però després de buscar sobre aquests, es va arribar a la conclusió que possiblement la [raspberry](#) no estava preparada per a la codificació de vídeos en temps real, ni en aquests formats, i si fos possible, el seu rendiment seria molt baix.

Decisió final

Veient la quantitat de problemes trobats, es va decidir no emprar més temps en buscar altres solucions, i quedar-nos en l'execució en local i visió del vídeo mitjançant una aplicació multimèdia instal·lada al nostre ordinador. Es va decidir això, ja que a la càmera únicament hi tindran accés els empleats i gerents de la botiga, i entenem que en el seu dispositiu se'ls podrà instal·lar sense cap complicació, ja que estem parlant de menys de 5 persones.

S'inclou el problema en l'apartat de millores futures del treball, ja que es creu que ha de ser possible, i també per a la motivació que han generat els entrebancs sorgits.

Finalment es pot veure el missatge previ a l'execució de la comanda que ens mostra el vídeo en local a la figura 8.11 i una captura del vídeo reproduint-se en local a la figura 8.12.

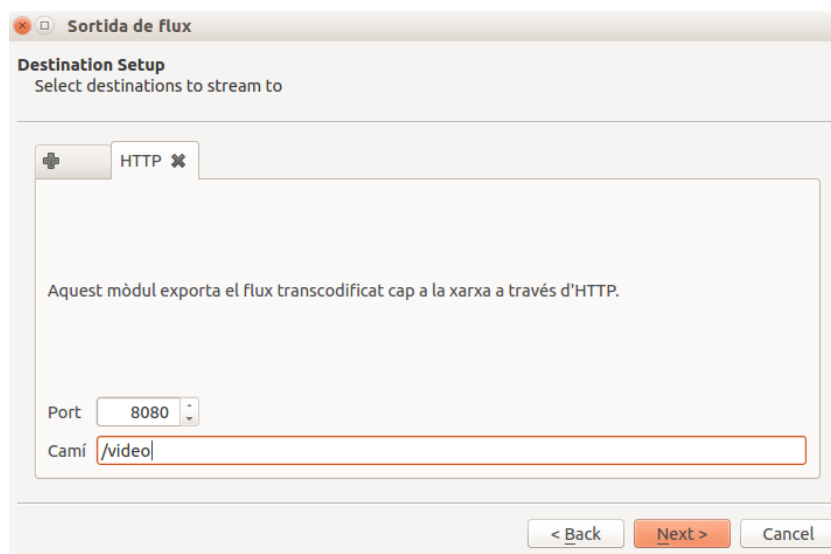


Figura 8.8: Captura del pas 2 de la transmissió del vídeo en protocol HTTP

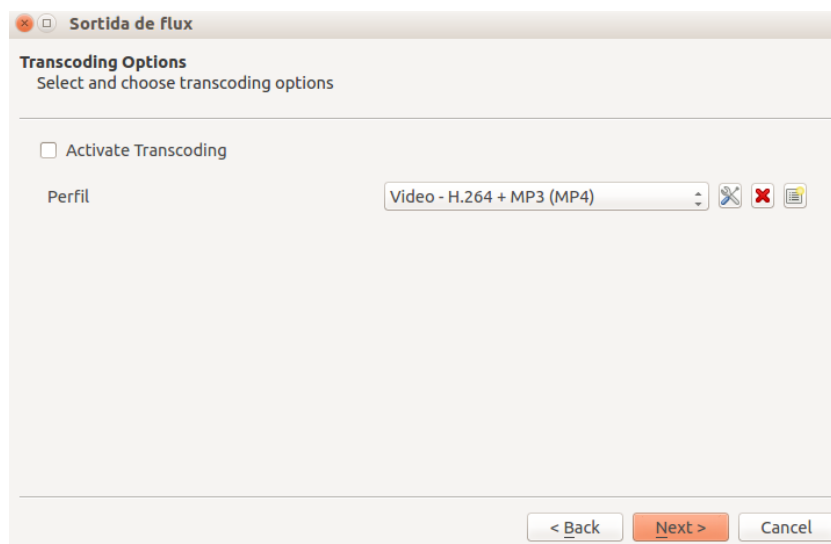


Figura 8.9: Captura del pas 3 de la transmissió del vídeo en protocol HTTP

```

pi@raspberrypi ~ $ cvlc rtsp://Quitxalla:camera-2016@192.168.1.102/videoMain --sout '#tr
anscode{vcodec=theo,vb=400,width=368,heigh=208,acodec=none}:http{mux=ogg,dst:8082/webcam
.ogg}'
VLC media player 2.0.3 Twoflower (revision 2.0.2-93-g77aa89e)
[0x11875b0] inhibit interface error: Failed to connect to the D-Bus session daemon: Unab
le to autolaunch a dbus-daemon without a $DISPLAY for X11 sensors telecitors, reflective.py: . . .
[0x11875b0] main interface error: no suitable interface module
[0x11882b0] main interface error: no suitable interface module
[0x11678f0] main libvlc error: interface "globalhotkeys,none" initialization failed
[0x11882b0] dummy interface: using the dummy interface module...
[0x11f3250] mux_ogg mux: Open
[0x11f3ce8] stream_out_transcode stream out error: cannot find audio encoder (module:any
fourcc:none). Take a look few lines earlier to see possible reason.
[0x11f3ce8] stream_out_transcode stream out error: cannot create audio chain
[0x11ae440] main decoder error: cannot create packetizer output (mlaw)
MultiFramedRTPSource::doGetNextFrame1(): The total received frame size exceeds the clien
t's buffer size (65536). 11090 bytes of trailing data will be dropped!
[h264 @ 0x11e0060] Invalid level prefix
[h264 @ 0x11e0060] error while decoding MB 20 18
[h264 @ 0x11e0060] concealing 229 DC, 229 AC, 229 MV errors

```

Figura 8.10: Errors en l'execució de la comanda per transmetre el vídeo a la raspberry.

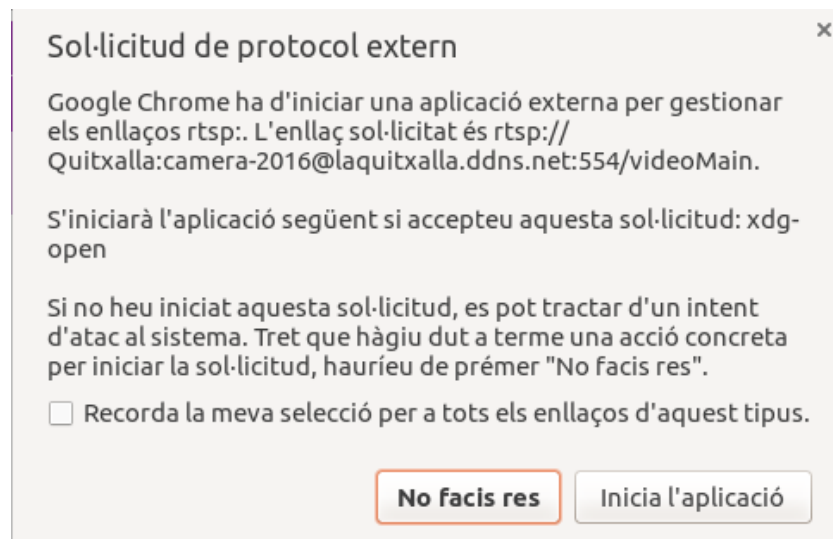


Figura 8.11: Missatge previ a l'execució de la comanda per veure el vídeo en local.

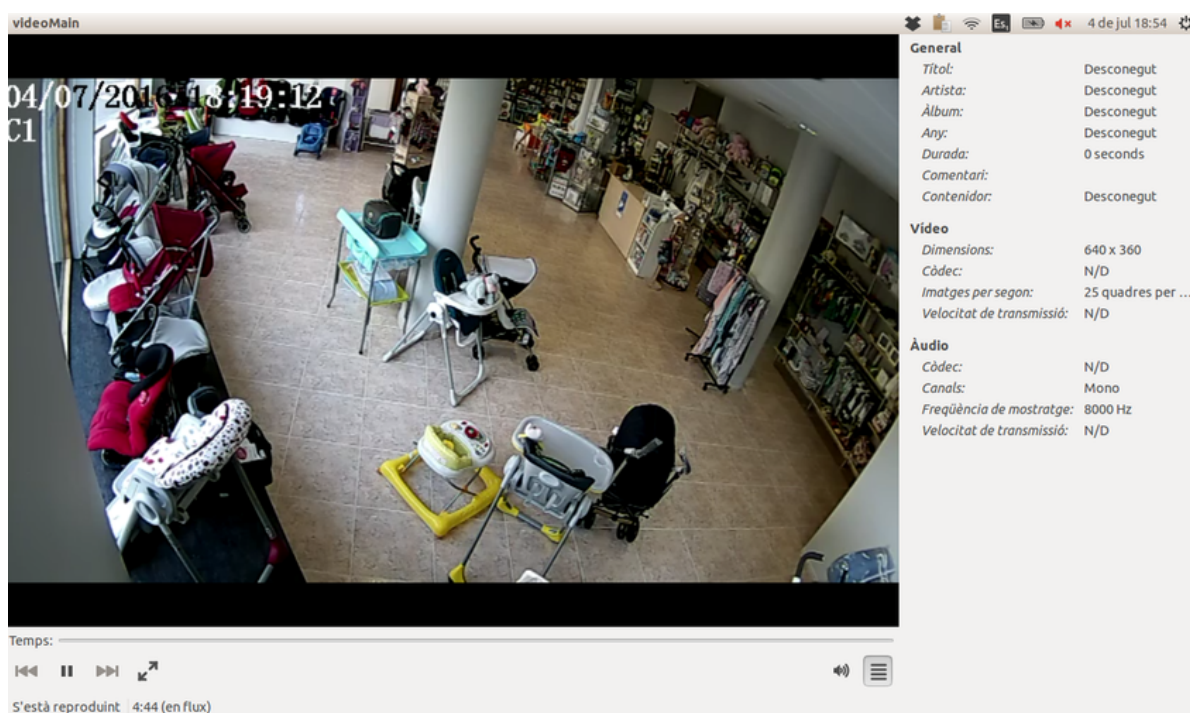


Figura 8.12: Captura de pantalla de la visió de la càmera des de l'aplicació local.

9 Conclusions

Fent un treball, sempre s'aprèn, fent un projecte encara més, però no hi ha millor manera d'aprendre que fent el treball final de grau del que més t'agrada.

Com a valoració final d'aquest projecte, es pot dir que s'han aconseguit tots els objectius proposats a l'inici. La creació de la pàgina web, la instal·lació dels sensors i la càmera, i l'obtenció de les dades i integració d'aquestes a l'entorn web. Per arribar al punt d'obrir la pàgina web i que tot es mostri amb un estil correcte, que totes les pàgines serveixin el contingut que toca, i que en l'apartat d'administració es vegi la informació ordenada, clara i precisa, ha requerit moltes hores de paciència i dedicació.

Aquest projecte m'ha ensenyat que bona part de les hores dedicades no han estat en la implementació, disseny o muntatge de qualsevol part del projecte, sinó en la recerca d'informació prèvia i la resolució d'errors posterior. Abans de muntar qualsevol sensor, s'havia d'estudiar com havia de ser connectat, així com el software necessari perquè aquest funcionés. També calia agafar paper i boli i pensar primer en el que després acabaria essent plasmat en un programa o fitxer corresponent a una vista de la pàgina web.

En el present projecte he treballat amb l'electrònica, uns dels punts que més respecte li tenia inicialment, ja que era la part del projecte que menys m'agradava. Després de barallar-mi moltes hores, he vist que si s'estudia bé el que es vol fer, i com fer-ho, no és tant complicat. Per a mi el muntatge de sensors requereix una lectura acurada del [datasheet](#), i pensar realment el que volem aconseguir abans de posar-nos en la pràctica. Si es segueixen aquests dos passos, a l'hora del muntatge tens una idea clara del que estàs fent i tens més recursos si sorgeix algun problema.

Web2py ha estat el framework amb el qual s'ha treballat. Inicialment pensava que ho havia d'haver escollit web2py, i haver programat la web amb django, que ja el coneixia. Al cap d'unes setmanes d'intensa dedicació a llegir la documentació i entendre com funcionava, i com havia de treballar amb aquest framework, de veritat que no me'n vaig penedir. Web2py m'ha ofert moltes facilitats a l'hora d'incorporar la informació dels sensors a la web, també a l'hora de tractar amb la base de dades, i sobretot amb la part d'autenticació ja només vaig haver de modificar el codi que web2py proporciona.

Cal dir també que he treballat amb molts llenguatges diferents. Al tenir el tema de la pàgina web, i del software que em permet interactuar amb els sensors, ha estat necessari adaptar-me en cada moment amb el llenguatge requerit. El fet de no fer un projecte d'una sola cosa, m'ha obligat a aprendre més llenguatges i no només un de sol, fet valorable de forma molt positiva.

La memòria del treball, ha estat escrita amb \LaTeX . Vui dir, una vegada més, que les facilitats que \LaTeX et dona i la diversitat de coses que et permet fer, és totalment recomanable la uti-

lització d'aquesta eina per a tothom qui hagi de fer una documentació extensa i sobretot d'un àmbit d'enginyeria o tècnic. \LaTeX et permet estructurar la memòria en diferents arxius per tal de poder treballar millor i tenir-ho tot organitzat. També et crea els índexs i la bibliografia de forma automàtica.

Una vegada més, vull dir que estic molt orgullós del treball realitzat, pel fet que no m'he basat o investigat només en un sol aspecte. El projecte tenia dos fronts molt importants com eren els sensors i la web, i altres de secundaris, però no menys importants, com la decisió dels sensors, frameworks, disseny de la web, ubicació dels sensors, comunicació entre raspberry... Gràcies a haver treballat amb tantes coses i tan diferents unes de les altres, ha obligat a formar-me i aprendre una mica de tot, portant-me a ser capaç de decidir, programar, muntar, configurar i implementar un resultat com el que he obtingut. He creat una pàgina web i he implementat un sistema de sensors en una botiga, i els he fusionat en un sol projecte. Fet totalment satisfactori per a mi.

Durant el grau, a part de tota la teoria i pràctica se'ns ha intentat formar com a enginyers, com a persones capaces de pensar i desenvolupar-nos davant d'un problema, i això crec que és el que he demostrat fent aquest projecte. En diversos punts del treball pensava que em costaria sortir-me'n i tirar endavant, però pensant una mica més del compte i posant un punt de pausa m'ha acabat sortint.

Ja per anar acabant, m'agradaria dir que la descripció detallada dels mòduls, així com l'explicació de tots els passos seguits per a implementar tot el projecte final, ha de servir a tot aquell que vulgui fer del seu negoci, casa o local, un lloc sensoritzat i totalment controlat a través d'internet. Per a fer-ho es poden escollir els mòduls que es desitgin i connectar-los seguint els passos explicats en la memòria.

Ara sí només cal afegir que aquest projecte no és un projecte acabat. Els objectius s'han complert, però la feina per fer no s'acaba. A continuació es citen una quantitat de millores i idees per ampliar el que fins aquí ha estat el meu projecte final de grau. I ja podeu estar segurs que moltes de les millores seran implementades.

10 Futures millores

En tot projecte es proposen unes futures millores, o bé formes d'ampliar el projecte realitzat. Aquest projecte a diferència de la resta, moltes de les següents millores seran implementades, ja que el projecte resta instal·lat en l'establiment, i per tant s'ha de fer un manteniment i millores constant.

A continuació se citen un seguit de millores atractives per a l'ampliació del sistema i robustesa del codi.

10.1 Millores en la pàgina web

- Fer que la pàgina web sigui totalment [responsive](#) per a tot tipus de dispositius mòbils, ja que ha estat un aspecte que tot i que web2py, bootstrap i materialize ho incorporen no s'ha tingut una cura especial, per tant s'entén que hi hagi possibles punts en la part d'administració que no es vegin adequadament.
- La velocitat de càrrega de les imatges de la pàgina web és lenta, tot i que les imatges són de baix pes, no hi ha cap [caché](#) que les guardi i serveixi de forma més ràpida. [Varnish](#) és una bona opció per a millorar aquest aspecte. Un altre punt podria ser disposar d'imatges de diferents mides i servir una o altra en funció de la mida màxima en la qual es veurà.
- El tercer i punt més atractiu, serà el de fer que es pugui vendre per internet. En l'apartat de productes, actualment simplement en veiem informació, per tant caldria gestionar tot el tema de la seguretat, mètodes de pagament, etc. per fer de la web un punt de venda en línia.

10.2 Millores en els sensors

- Aprofitant que s'han instal·lat i configurat els sensors de temperatura, una bona millora seria aprofitar-los per controlar la calefacció i l'aire condicionat amb un relé quan els sensors indiquessin que és supera els límits de temperatura establerts.
- Es podria aprofundir també en el tema de les gràfiques o registre de la informació dels sensors. Fer gràfiques diàries, mensuals o anuals, seria una bona idea per a tenir estadístiques molt valuoses per a la botiga. Si parlem de gràfiques de temperatura, i dels clients que entren a la botiga durant tot l'any, podríem conèixer l'afluència de gent a la botiga en funció de la calor o fred, o simplement quina època de l'any hi ha més o menys clients.
- La detecció de moviment a través de la càmera, és una característica que ja permet la nostra càmera. Per tant establir algun tipus d'alarma o missatge quant es detecti moviment, seria una forma d'augmentar la seguretat.

- A partir del programa que disposem, que ens mostra la quantitat de clients que hi ha a la botiga, es podria enviar un missatge utilitzant el bot de [telegram](#), quant hi hagués més de X persones, per avisar a la gerent que falta personal.

Bibliografia

- [1] Amazon. *Raspberry Pi 3 Modelo B (1,2 GHz Quad-core ARM Cortex-A53, 1GB RAM, USB 2.0): Amazon.es: Informática.* https://www.amazon.es/gp/product/B01CD5VC92/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1. (Cons. 16-03-2016).
- [2] *Bootstrap · The world's most popular mobile-first and responsive front-end framework.* <http://getbootstrap.com/>. (Cons. 07-06-2016).
- [3] *Commutador o switch - Xarxes informàtiques.* <https://goo.gl/DMEX1v>. (Cons. 06-05-2016).
- [4] DigiKey. *CY-191A-Z-Y Panasonic Industrial Automation Sales — Sensores y transductores — DigiKey.* <http://www.digikey.com/product-detail/es/panasonic-industrial-automation-sales/CY-191A-Z-Y/1110-1460-ND/3898993>. (Cons. 15-03-2016).
- [5] *Documentation - Materialize.* <http://materializecss.com/>. (Cons. 07-06-2016).
- [6] *Download — Automation Controls — Industrial Devices — Panasonic.* http://www3.panasonic.biz/ac/e_download/fasys/sensor/photoelectric/catalog/cy-100_e_cata.pdf. (Cons. 15-03-2016).
- [7] *DS18B20 Temperature Sensor with Raspberry Pi - Raspberry Pi.* <http://www.reuk.co.uk/DS18B20-Temperature-Sensor-with-Raspberry-Pi.htm>. (Accessed on 14-04-2016).
- [8] Farnell. *RASPBERRYPI-2-MODB-1GB. - RASPBERRY-PI - SBC, RASPBERRY PI 2, MODELO B, 1GB RAM — Farnell element14 España.* <http://goo.gl/R9lQ9L>. (Cons. 25-04-2016).
- [9] Foscam. *Foscam C1.* <http://foscam.com/C1.html>. (Cons. 15-03-2016).
- [10] *Framework.pdf.* http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf. (Cons. 07-06-2016).
- [11] *GPIO - Wikipedia, la enciclopedia libre.* <https://es.wikipedia.org/wiki/GPIO>. (Cons. 26-05-2016).
- [12] *GPIO: Raspberry Pi Models A and B - Raspberry Pi Documentation.* <https://www.raspberrypi.org/documentation/usage/gpio/>. (Cons. 25-05-2016).
- [13] *Laravel - The PHP Framework For Web Artisans.* <https://laravel.com/>. (Cons. 07-06-2016).
- [14] Simon Monk. *Overview — Adafruit's Raspberry Pi Lesson 11. DS18B20 Temperature Sensing — Adafruit Learning System.* <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/overview>. (Cons. 25-2-2016).
- [15] *Raspberry Pi - Viquipèdia, l'enciclopèdia lliure.* https://ca.wikipedia.org/wiki/Raspberry_Pi. (Cons. 07-04-2016).

- [16] RS-Online. *0900766b8078b130.pdf*. <http://docs-europe.electrocomponents.com/webdocs/078b/0900766b8078b130.pdf>. (Cons. 25-04-2016).
- [17] RS-online. *Raspberry Pi 3 Model B — Raspberry Pi 3 Model B SBC — Raspberry Pi*. <http://goo.gl/6jt1zY>. (Cons. 25-04-2016).
- [18] *Simple Guide to the RPi GPIO Header and Pins*. <http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>. (Cons. 05-05-2016).
- [19] Optek Technology. *OPB715-716-717-718_A4.pub*. <http://optekinc.com/datasheets/OPB715-716-717-718.PDF>. (Cons. 15-03-2016).
- [20] *The Web framework for perfectionists with deadlines — Django*. <https://www.djangoproject.com/>. (Cons. 07-06-2016).
- [21] *web2py vs. Django vs. Laravel comparison — vsChart.com*. <http://vschart.com/compare/web2py/vs/django-framework/vs/laravel>. (Cons. 07-06-2016).
- [22] *web2py_vs_others.pdf*. http://www.web2py.com/examples/static/web2py_vs_others.pdf. (Cons. 07-06-2016).
- [23] Andrew Williams. *Raspberry Pi 3 vs Pi 2: What's the difference?* <http://goo.gl/yMkXa8>. (Cons. 15-03-2016).
- [24] *www.web2py.com*. <http://www.web2py.com/>. (Cons. 07-06-2016).